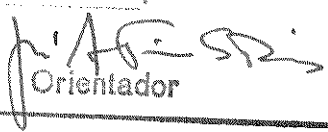


GOVERNO DO ESTADO DE SÃO PAULO

UNIVERSIDADE ESTADUAL DE CAMPINAS

Este exemplar corresponde à redação final da tese
defendida por LAÉRCIO CALDEIRA
Julgadora em 10.02.84 aprovada pela Comissão

Orientador

FACULDADE DE ENGENHARIA ELÉTRICA

PUBLICAÇÃO

FEE

13.081 - CAMPINAS - SP

Laércio Caldeira

127

Engenheiro Eletricista, EFEI, 1.976.

Mestre em Engenharia, EFEI, 1.979.

BLOCOS CMOS DE ALTA PERFORMANCE PARA APLICAÇÕES EM VLSI

Tese apresentada à Faculdade de Engenharia Elétrica da UNICAMP como requisito parcial à obtenção do título de Doutor em Engenharia Elétrica.

Orientador: Prof. Dr. José Antônio Siqueira Dias

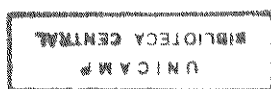
Universidade Estadual de Campinas

Faculdade de Engenharia Elétrica

Departamento de Eletrônica e Microeletrônica

Laboratório de Eletrônica e Dispositivos

Dezembro de 1.993



*Aos meus pais,
Pedro e Lúzia
e minha mulher,
Cida,
dedico.*

Este trabalho contou com o apoio técnico e financeiro da Telecomunicações Brasileiras SA, através do Convênio EFEI-DON / CPqD-Telebrás (Contratos 226/87, 276/88, 358/89 e 455/91) e do Centro Tecnológico para Informática, através do Projeto Multi-Usuário Brasileiro.

AGRADECIMENTOS

Nesta oportunidade ímpar, que registra um marco em nosso desenvolvimento profissional, apraz-nos agradecer ao **Prof. José Antônio Siqueira Dias** que, desde o nosso primeiro contato, premiou-nos com a sua confiança, capacidade, apoio irrestrito e principalmente, com a sua amizade. A nossa mais sincera e justa consideração por viabilizar este trabalho.

Não seria justo deixarmos de agradecer também ao **Prof. Ozeas Valente de Avilez Filho**, da UNICAMP, pelas diretivas que ajudaram a nortear os nossos trabalhos iniciais e aos engenheiros **Robson Luiz Moreno**, **Adriano Marques Pereira**, **Eduardo Ferreira Rodrigues**, **Janete Mouallem**, **Marília dos Santos** e **Marcos Aurélio de Oliveira**, pelo apoio durante os projetos dos blocos e suas caracterizações.

RESUMO

Este trabalho integra o projeto de três blocos CMOS digitais de alta performance idealizados para incorporarem CIs VLSI, onde se buscou o melhor compromisso área/velocidade. Um dos blocos, o do Conversor A/D de 8 Bits, desenvolvido em arquitetura "flash" modificada, permite taxas de conversão superiores a 1 MHz adequando-o à grande maioria dos sistemas digitais. Os demais blocos, os Multiplicadores Paralelo e Série, foram desenvolvidos com base no Algoritmo de Booth e, portanto, exibem excelente desempenho trabalhando com palavras de até 64 bits. Projetou-se um Multiplicador Paralelo de 4 Bits operando com taxas de até 35 MHz e um Multiplicador Série de 8 Bits operando até 2,6 MHz.

ABSTRACT

This work compiles the design of three high performance CMOS digital circuits blocks where the best area/speed relationship was required. Those circuits are intended to be implemented in VLSI circuits. One of the blocks, the 8 bits A/D Converter, was implemented using a modified flash architecture, which allows conversion ratio above 1 MHz, suitable to the most of the digital systems. The other blocks, a parallel and a Serial Multipliers were developed based on the Booth Algorithm, wich has higher performance with words up to 64 bits. The 4 bits parallel multiplier designed works up to 35 MHz and the 8 bits serial multiplier works up to 2,6 MHz.

SUMÁRIO

1. APRESENTAÇÃO

1.1 Breve Histórico	1.1
1.2 A Decisão	1.3

2. O CONVERSOR A/D DE OITO BITS

2.1 Objetivos	2.1
2.2 Os Conversores A/D	2.2
2.2.1 Considerações sobre Conversores	2.2
A) O Conversor Escada	2.2
B) O Conversor por Aproximações Sucessivas	2.4
C) O Conversor de Rampa Simples	2.5
D) O Conversor de Rampa Dupla	2.7
E) O Conversor "Flash"	2.9
2.2.2 O Conversor A/D Proposto	2.12
2.3 Os Comparadores	2.18
2.3.1 Configurações de Comparadores	2.21
A) Comparador tipo Cascata de Inversores	2.21
B) Comparador com Amplificador Diferencial	2.25
2.3.2 O Comparador Proposto no Projeto	2.26
2.4 Projeto dos Blocos do Conversor	2.32
2.4.1 O Comparador	2.32
A) Os Transistores-Chave	2.32
B) O Estágio de Entrada	2.33
C) O Estágio Amplificador	2.36
D) O Estágio "Latch"	2.37
2.4.2 Os Resistores de Referência	2.42

2.4.3 O Codificador de 15 Entradas e 4 Saídas	2.42
2.4.4 O Gerador das Fases de "Clock" e Sinais de Controle	2.46
2.4.5 O Multiplexador	2.55
2.4.6 Os Registradores de Saída	2.59
2.4.7 O "Pad" Analógico de Saída	2.60
2.4.8 "Sample-and-Hold" e a Referência $V_{dd}/2$	2.61
2.5 Considerações sobre o "Layout" do Conversor	2.62
2.5.1 O Comparador	2.62
A) Cálculo das Dimensões dos Capacitores	2.62
B) Cálculo dos Resistores de Referência	2.63
2.5.2 Os Outros Blocos	2.64
2.5.3 As Estruturas de Teste	2.64
2.6 Resultados e Conclusões	2.65
 3. OS MULTIPLICADORES DIGITAIS	
3.1 Considerações sobre Multiplicadores	3.1
3.2 O Multiplicador Série	3.3
3.3 O Multiplicador Série-Paralelo	3.5
3.4 O Multiplicador Paralelo	3.6
3.5 O Algoritmo de Booth	3.12
3.6 O Algoritmo de Booth Modificado	3.13
3.6.1 As Funcionalidades Necessárias no Bloco	3.18
3.6.2 O Problema da Extensão do Sinal	3.21
3.6.3 O Método do Sinal Gerado	3.22
 4. O MULTIPLICADOR PARALELO	
4.1 O Circuito C3	4.2
4.2 O Circuito C1	4.3
4.3 O Circuito C2	4.3
4.4 Os Circuitos Somadores	4.5

4.4.1 O Circuito do Meio-Somador (MS)	4.5
4.4.2 O Circuito do Somador Completo (SC)	4.6
4.4.3 O Circuito do Somador Completo Especial (SCE)	4.6
4.5 O Circuito do "Carry Look-Ahead" (CLA)	4.6
4.6 Portas Lógicas Especiais	4.7
4.7 O Diagrama em Blocos do Multiplicador	4.8
4.8 As Simulações	4.9
4.8.1 A Simulação Elétrica	4.9
4.8.2 Análise da Frequência de Operação	4.16
4.8.3 A Simulação Lógica	4.18
4.9 A Elaboração do "Layout" do Multiplicador	4.18
4.10 Resultados e Conclusões	4.19

5. A IMPLEMENTAÇÃO DO MULTIPLICADOR SÉRIE

5.1 Considerações Gerais	5.1
5.2 O Multiplicador Recodificado	5.3
5.2.1 Recodificação Utilizando o Algoritmo de Booth Modificado	5.5
5.3 O Projeto do Multiplicador	5.6
5.3.1 A Célula Básica CELBA	5.8
5.3.2 Os Registradores de Deslocamento	5.16
5.3.3 A Unidade de Controle	5.19
5.4 A Simulação do Multiplicador	5.30
5.4.1 A Simulação Elétrica	5.30
5.4.2 A Simulação Lógica	5.37
5.5 O "Layout" do Multiplicador	5.38
5.6 Resultados e Conclusões	5.39

6. CONCLUSÕES FINAIS

BIBLIOGRAFIA CITADA E RECOMENDADA

CAPÍTULO 1

APRESENTAÇÃO

1.1 BREVE HISTÓRICO

Em 87, foi celebrado um convênio entre o CPqD /TELEBRÁS e a Escola Federal de Engenharia de Itajubá, com o objetivo de se implantar em Itajubá um programa voltado ao projeto de circuitos integrados dedicados, que pudesse apoiar o desenvolvimento dos sistemas digitais de interesse daquela empresa.

Para se ajustar a linha de ação do programa, considerou-se que:

- o grau de complexidade dos projetos de circuitos digitais vem crescendo de forma assombrosa nos últimos anos;

- a tendência mundial de se projetar "single-chip systems" é cada vez mais evidente. Como comprovação, hoje já se anuncia o "PC descartável" com a utilização do circuito integrado PC/chip da "Chips and Technologies" [22], que substitui todos os integrados usados em um sistema PC 386. Outra comprovação é a certeza de que esta inovação surpreendente noticiada estará obsoleta em pouquíssimo tempo.

Dentro deste contexto mundial, acreditou-se que a melhor estratégia a ser adotada para se projetar circuitos VLSI seria a de se utilizar, cada vez mais, blocos previamente projetados e caracterizados, para permitir que a integração de uma sistema pudesse ser realizada de forma rápida, confiável e com o menor custo possível.

A estratégia proposta na época acabou se mostrando correta pois coincide hoje com a de um tema de muita importância no cenário nacional. Basta lembrar que em 92 foi criado o programa nacional de "Biblioteca de Células", que visa dotar o País de um conjunto de blocos de alta performance para facilitar o projeto de circuitos VLSI de alta complexidade.

Com esta visão ajustada no convênio, o trabalho foi iniciado com o objetivo de se desenvolver alguns blocos de alta performance, em tecnologia CMOS, que pudessem ser utilizados em projetos de circuitos integrados VLSI, como parte de CIs de alta complexidade.

O trabalho contou com o suporte financeiro do CPqD/Telebrás e a maioria dos blocos escolhidos para serem projetados e fabricados (usando o Projeto Multi-Usuário Brasileiro) eram de interesse da Telebrás, que vislumbrava uma aplicação destes blocos como parte de circuitos integrados dedicados para o Sistema RDSI.

1.2 A DECISÃO

Embora, no âmbito do convênio, tenham sido projetados e testados vários blocos, para este trabalho de doutorado selecionou-se apenas alguns dos circuitos, por serem os que apresentavam o maior número de inovações em arquitetura e técnicas de projeto além de apresentarem um elevado grau de dificuldade.

Os blocos escolhidos para serem apresentados foram:

- Conversor A/D de 8 Bits;
- Multiplicador Digital Série;
- Multiplicador Digital Paralelo.

As especificações de cada um dos blocos será apresentada quando da descrição dos seus respectivos projetos, feita nos próximos capítulos.

CAPÍTULO 2

O CONVERSOR A/D DE OITO BITS

Este capítulo trata do projeto de um conversor análogo-digital de oito bits, desenvolvido para trabalhar com sinais de frequência de vídeo, ou seja, com taxas de amostragem superiores a 1 MHz.

2.1 OBJETIVOS

O projeto em foco nasceu da necessidade de se dispor de conversores capazes de acompanhar o desenvolvimento atingido pela área de processamento digital de sinais. Esta área é de vital importância no setor de telecomunicações, particularmente no campo do processamento digital de imagens.

No decorrer do texto, além do método de conversão A/D adotado na execução do projeto, outros serão ilustrados. Adicionalmente, serão apresentadas duas estruturas básicas de comparadores juntamente com os problemas ligados à comparação de pequenas diferenças de tensão.

Objetivamente, o projeto deste conversor busca o estudo de estruturas para a implementação de conversores que operem segundo as seguintes exigências:

- tecnologia "flash";
- resolução de 8 bits;
- operação com frequências de amostragem superiores a 1 MHz;
- devem ser capazes de resolver diferenças de até $1/4$ do LSB;
- faixa dinâmica de 0 a 5V;
- tensões de alimentação e de saída TTL compatíveis;
- ocupem pequena área de integração.

Para a caracterização das estruturas e do conversor projetado, foi prevista a difusão de estruturas de teste e do conversor no PMU CMOS 5. Os resultados deverão ampliar os estudos e conduzir a novos projetos.

2.2 OS CONVERSORES A/D

2.2.1 CONSIDERAÇÕES SOBRE CONVERSORES

Atualmente, os conversores A/D têm assumido um papel de relevante importância na área de comunicação devido às vantagens do processamento de sinais

digitais sobre o de sinais analógicos.

Como função da velocidade de conversão e da precisão, diferentes técnicas de conversão análogo/digital vêm sendo desenvolvidas. A Fig. 2.1 traz uma classificação não muito rigorosa dos conversores A/D segundo as técnicas mais comuns de conversão [20]. Segue-se uma rápida apresentação dos mesmos.

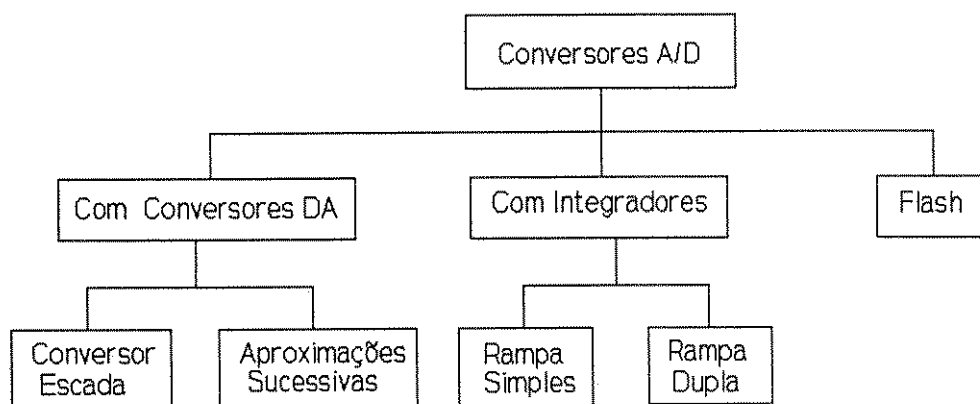


Fig. 2.1 Classificação dos Conversores A/D

A - O CONVERSOR ESCADA

O conversor escada, que pode ser visto na Fig. 2.2, é o mais simples conversor A/D e também o menos usado. Neste circuito, um sinal de "clock" é aplicado em um contador cuja saída é levada a um conversor D/A. Este conversor transforma a contagem em uma tensão analógica proporcional que é aplicada na entrada de um comparador, que faz a comparação entre este sinal e a entrada analógica.

A saída do comparador dá uma indicação digital de se a entrada analógica

é maior ou menor que a saída do conversor D/A. Assim, aplicando-se a saída do comparador na entrada de habilitação do contador (porta E), pode-se inibir a contagem quando os dois sinais forem iguais. A contagem no contador, quando este for desabilitado, será o valor digital convertido da entrada analógica.

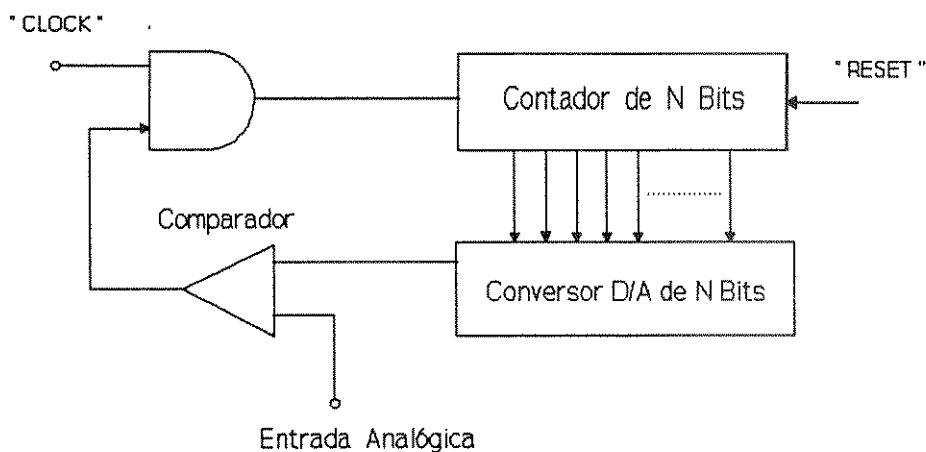


Fig. 2.2 O Conversor Escada

O conversor escada apresenta um tempo de conversão variável e pode chegar a 2^n períodos de "clock", onde n é o número de bits do contador.

B - O CONVERSOR POR APROXIMAÇÕES SUCESSIVAS

O conversor por aproximações sucessivas é mais rápido que o conversor escada e seu tempo de conversão é constante e igual a $n+1$ pulsos de clock (n é o número de bits do conversor). O seu algoritmo de conversão é mais complexo e seu fluxograma pode ser visto na Fig. 2.3, que também traz o seu diagrama em blocos.

O contador em anel de $n+2$ bits ajusta inicialmente o bit mais

significativo do registro de saída em 1. Através do comparador, a contagem presente no contador é comparada com a entrada analógica. Se a contagem for maior, no próximo pulso de "clock" o segundo bit do contador será feito igual a 1 e o mais significativo será zerado". Caso seja menor, o segundo bit será feito igual a 1 e se manterá o bit mais significativo.

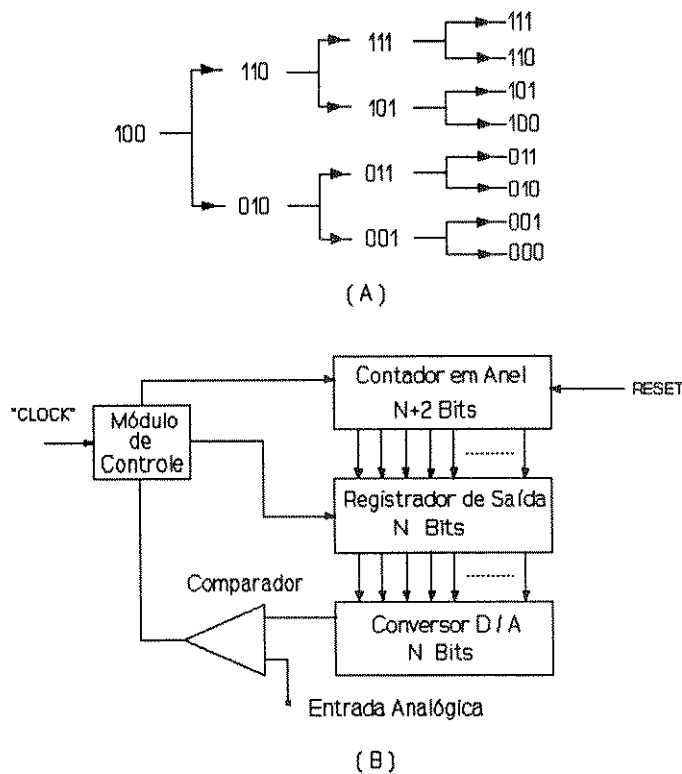


Fig. 2.3 Conversor por Aproximações Sucessivas

O procedimento acima é repetido para os bits menos significativos do contador até o seu último bit, conforme pode ser visto no fluxograma da Fig. 2.3a.

C - O CONVERSOR DE RAMPA SIMPLES

O conversor de rampa simples, que pode ser visto na Fig. 2.4, tem a propriedade de integrar o sinal de entrada tornando-se, portanto, imune aos ruídos usualmente presentes neste sinal.

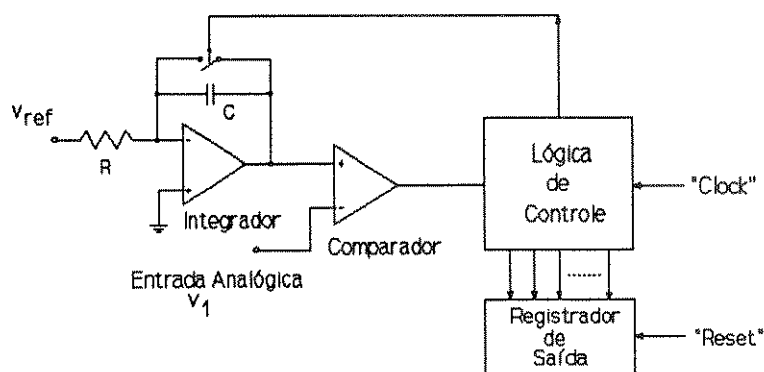


Fig. 2.4 Conversor de Rampa Simples

Durante o primeiro pulso de "clock", um contador interno à lógica de controle é zerado e o integrador começa a integrar a tensão de referência (V_{ref}). Enquanto a tensão da rampa for menor que a tensão de entrada, a saída do conversor manterá o contador habilitado.

Quando a tensão da rampa se tornar igual à tensão de entrada, o comparador desabilitará o contador e a contagem presente do contador será o valor digital correspondente à tensão analógica de entrada. Esta afirmativa pode ser confirmada pela equação:

$$V_{ent} = \frac{V_{ref} t_i}{RC} \quad (2-1)$$

onde:

V_{ent} = tensão na saída do integrador;

V_{ref} = tensão de referência;

R = resistência de integração;

C = capacitância de integração;

t_i = tempo de integração.

Além disso, tem-se que:

$$t = n \cdot T \quad (2-2)$$

onde:

t = intervalo de tempo necessário para que ocorram "n" pulsos de "clock".

T = período do pulso de "clock".

O contador é desabilitado quando V_{ent} é igual a entrada analógica V_1 .

Neste instante $t_i = t$ e, logo:

$$V_1 = \frac{V_{ref}}{RC} \cdot T \cdot n \quad (2-3)$$

Como V_{ref} , R , C e T são constantes, a tensão de entrada V_1 será proporcional ao número de pulsos de "clock" contados (n).

O principal problema deste tipo de conversor é a necessidade de se gerar uma rampa cuja inclinação seja precisa e invariante.

D) O CONVERSOR DE RAMPA DUPLA

O conversor de rampa dupla também integra o sinal de entrada e, por isto, é imune aos ruídos presentes neste sinal. A Fig. 2.5 mostra o diagrama em blocos de um conversor deste tipo.

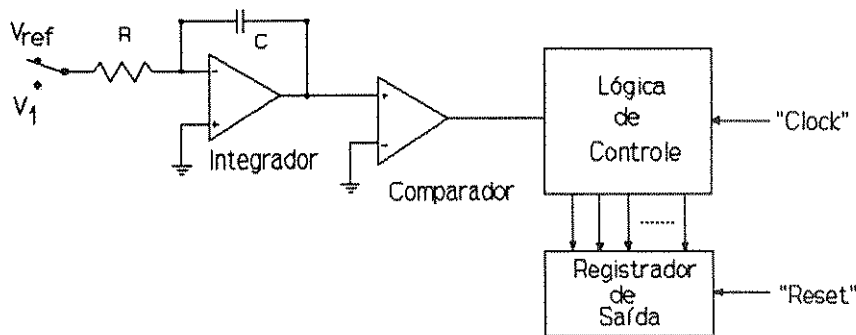


Fig. 2.5 - Conversor de Rampa Dupla

Inicialmente o sinal a ser convertido V_1 é aplicado na entrada do integrador durante um número definido de pulsos de "clock" (n). Assim sendo, se T é o período de "clock", o intervalo de tempo em que o sinal é aplicado na entrada do integrador será $n \cdot T$ e a sua tensão de saída:

$$V_{ent} = \frac{V_1}{RC} \cdot t \cdot n \quad (2-4)$$

onde:

R = resistência de integração;

C = capacitância de integração.

Em seguida, o integrador tem sua entrada ligada a uma tensão de referência (V_{ref}) de polaridade contrária à de V_1 , até que a sua saída retorne a zero. O número de períodos de "clock" (m) necessários para que a saída fique zerada

excita um contador interno da lógica de controle que ao término da contagem, transfere o valor para o registrador de saída. Assim,

$$V_{ent} = \frac{V_{ref}}{RC} \cdot T \cdot m \quad (2-5)$$

onde,

R = resistência de integração;

C = capacitância de integração;

T = período do pulso de clock.

Das Eqs. (2-4) e (2-5), tem-se:

$$m = n \cdot \frac{V_1}{V_{ref}} \quad (2-6)$$

Como V_{ref} e V_1 são constantes, o registrador de saída estará retendo um valor proporcional à tensão de entrada.

E) O CONVERSOR "FLASH"

O conversor "flash" é o mais rápido dos conversores apresentados e consiste de um conjunto de comparadores com referências de tensão geradas por um divisor resistivo e um codificador. A Fig. 2.6 apresenta o diagrama em blocos de um conversor "flash".

O codificador converte as saídas dos comparadores para o código binário equivalente à entrada analógica. Os resistores devem ser ajustados de forma a colocar a tensão de limiar dos comparadores no centro de um incremento.

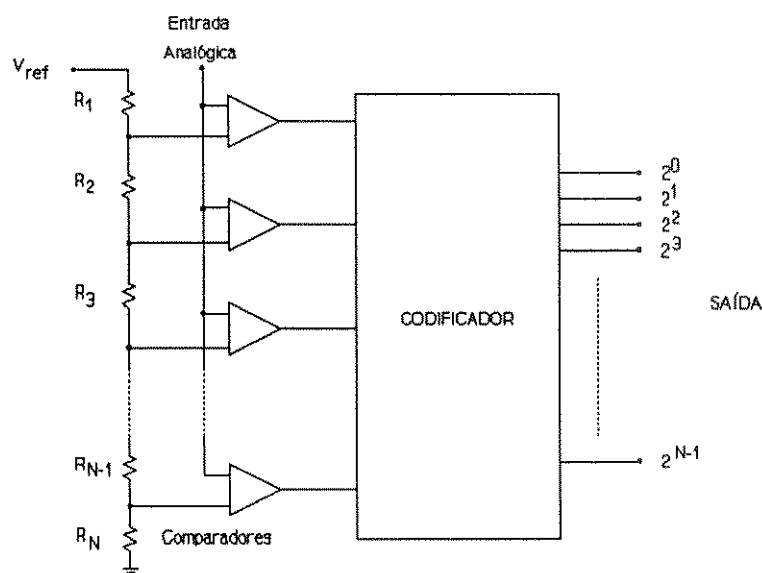


Fig. 2.6 O Conversor "Flash"

Para a construção de um conversor "flash" de n bits são necessários $2^n - 1$ comparadores. Isto implica em que conversores deste tipo e de grande precisão ocupam grandes áreas de silício.

Se a velocidade não for tão crítica, pode-se usar o circuito da Fig. 2.7 que é uma modificação do circuito anterior. Neste circuito somente um comparador é usado mas, em contrapartida, é usado um registrador de deslocamento.

A Fig. 2.8 mostra um circuito alternativo. Este circuito é menos veloz que o "flash" tradicional; entretanto, tem a vantagem de usar menos comparadores para uma mesma precisão pois trabalha em duas fases [20].

O circuito é constituído por dois conversores "flash" de $n/2$ bits operando segundo um método que será descrito adiante. A sua referência de tensão

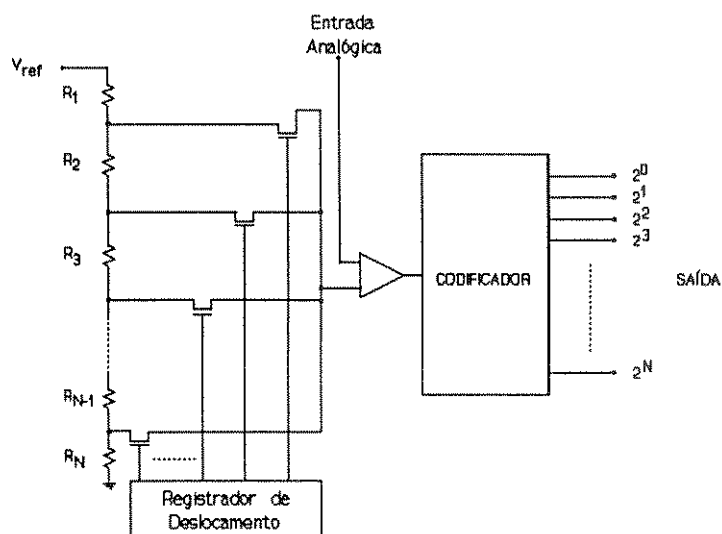


Fig. 2.7 - Conversor "Flash" com Registrador de Deslocamento

é formada por um grupo de "resistores grossos" e de um conjunto de "resistores finos" agrupados em paralelo com cada resistor grosso. Esta solução embora reduza o número de comparadores, faz crescer o número de resistores casados, não reduz a área final e conduz a um conversor mais lento, pelo que não foi adotada.

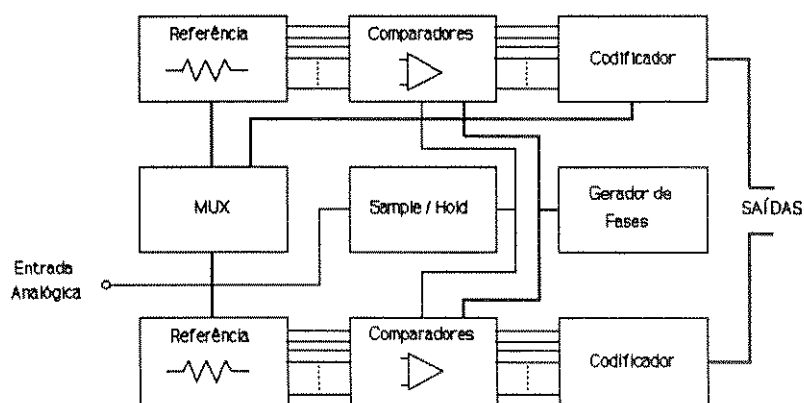


Fig. 2.8 Conversor A/D Alternativo

Entende-se como resistores grossos aqueles associados à conversão dos bits mais significativos e como resistores finos à dos bits menos significativos.

2.2.2 O CONVERSOR A/D PROPOSTO

O conversor usado no projeto é uma variação do conversor "flash". O seu diagrama pode ser visto na Fig. 2.9.

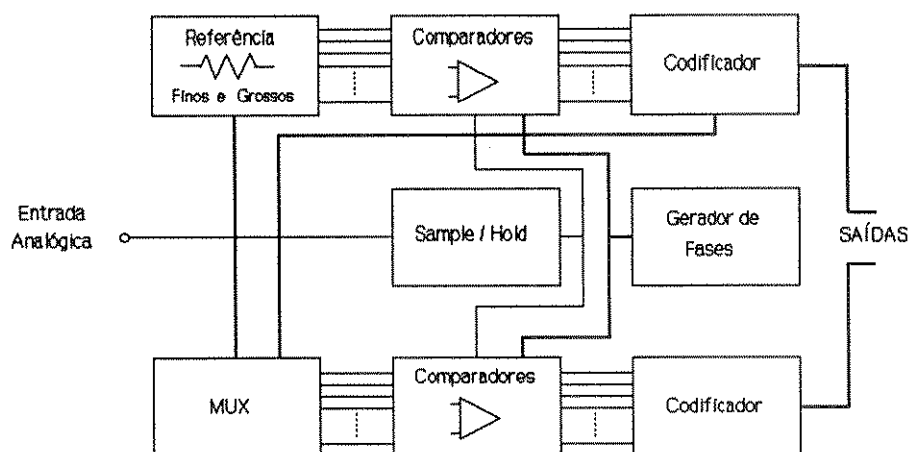


Fig. 2.9 - Conversor A/D Proposto

O conversor proposto é bastante semelhante ao conversor da Fig. 2.8. Sua principal diferença reside no fato de que um só conjunto de "resistores finos" é colocado em paralelo com um "resistor grosso" específico. Portanto, este conversor tem a vantagem de empregar um número de resistores casados bem menor que o da Fig. 2.8 ou do que o conversor "flash" tradicional.

O circuito da Fig. 2.9 é formado basicamente por dois blocos de comparadores (um "grosso" e um "fino"), dois conjuntos de resistores de referência (um "grosso" e um "fino"), dois codificadores, um gerador de fases, um circuito do tipo "sample-and-hold" e um multiplexador.

A Fig. 2.10 traz um diagrama em blocos que facilitará a explanação do funcionamento do circuito. O conjunto de resistores gera as referências de tensão

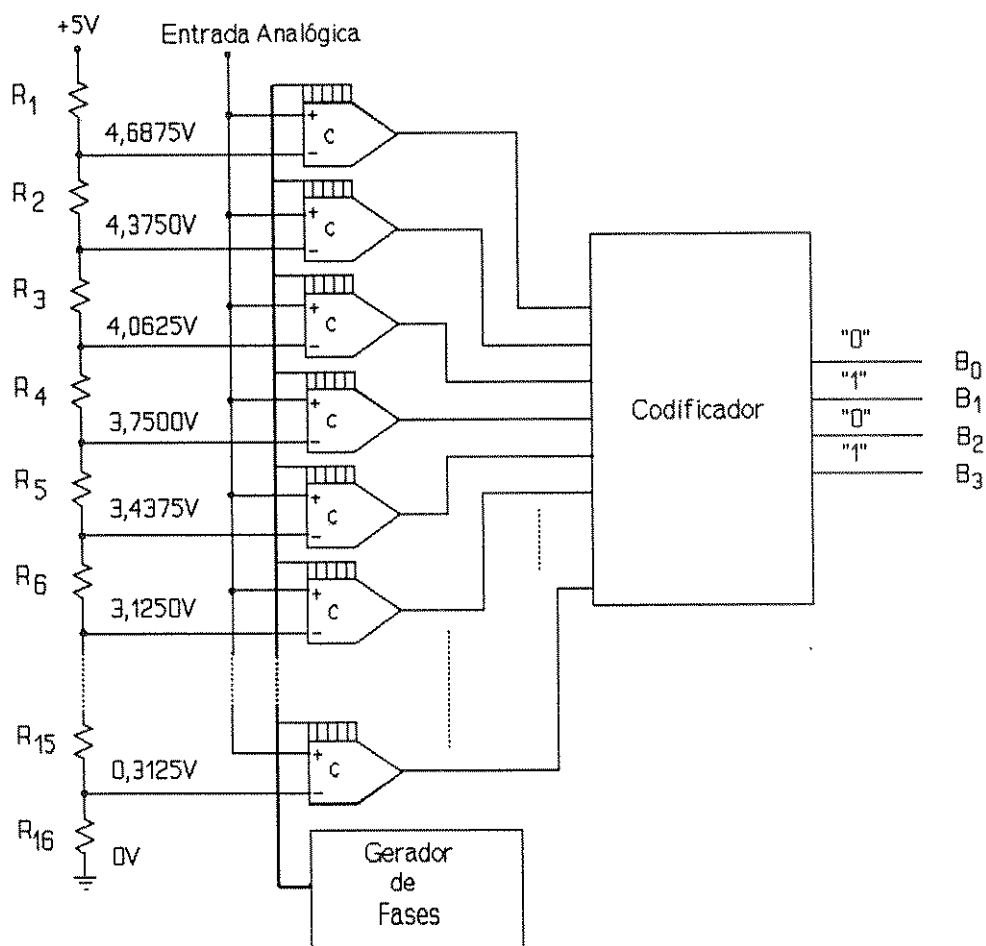


Fig. 2.10 O Conversor A/D

para os Comparadores. Estes, comparam a tensão de entrada analógica com a respectiva referência e geram uma saída em nível lógico "1" (indicando que a entrada é maior que a referência) ou "0" (indicando que a entrada é menor que a referência).

Todos comparadores, cujas tensões de referência forem menores que a da entrada analógica, exibirão uma saída 1 e as demais, uma saída 0. Na sequência, o codificador toma estas saídas e as converte para o código binário equivalente à entrada analógica.

Com o circuito da Fig. 2.10 ter-se-ia um conversor A/D de 4 bits. Como ilustração, indicou-se uma saída binária para uma entrada analógica de 3,5V e com os resistores alimentados como indicado.

Na saída do codificador da Fig. 2.9 é ligado um multiplexador que, de acordo com o valor da saída, liga um conjunto igual ao da Fig. 2.10 em paralelo com um dos resistores da referência. O resistor que terá um conjunto ligado em paralelo é aquele cujas extremidades estão ligadas, uma à referência maior que a entrada analógica e outra à referência menor que a entrada analógica. No exemplo da Fig. 2.10 seria o resistor R5.

O procedimento descrito permite a obtenção de um conversor de 8 bits a partir de dois conversores de 4 bits. O conjunto "grosso" converte os 4 bits mais significativos e o conjunto "fino" é colocado na faixa de 1 "LSB" do conjunto grosso e converte os 4 bits menos significativos.

A Fig. 2.11 mostra um diagrama mais detalhado do conversor proposto

e apresentado na Fig. 2.9. Ele é formado basicamente por dois blocos conversores e alguns circuitos periféricos. Os dois blocos conversores são formados, cada um, por um conjunto de resistores de referência, um conjunto de comparadores e um codificador.

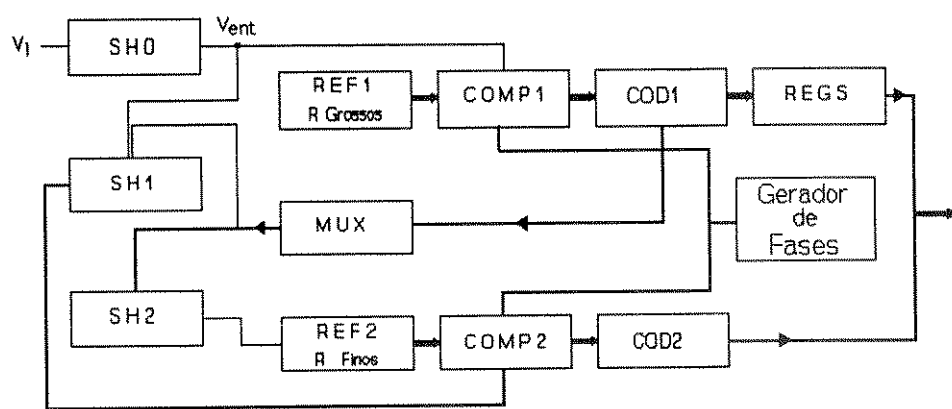


Fig. 2.11 O Conversor Proposto mais Detalhado

Os blocos operam da seguinte forma:

- Numa primeira fase é feita a auto-zeragem dos comparadores (a ser discutida posteriormente);
- Após o auto-zero, é efetuada a comparação entrada/referência;
- O resultado da comparação é fixado no "latch" do comparador;
- O codificador converte a saída dos comparadores para o código binário.

O circuito "sample-and-hold" de entrada "SH0" tem a função de amostrar o sinal de entrada durante a fase de auto-zero e retê-lo durante a fase de conversão. Além disto, o circuito ajuda a eliminar os ruídos presentes no sinal de entrada.

Os dois blocos de conversão operam em paralelo, convertendo os seus respectivos dados de entrada de forma simultânea. Entretanto, o segundo bloco necessita do resultado do primeiro (bits mais significativos) para executar a conversão dos bits menos significativos. Com isto, o segundo bloco trabalha sempre com uma amostra atrasada no tempo em relação ao primeiro bloco.

O fato dos blocos trabalharem defasados de um período de amostragem, gera a necessidade de se armazenar o resultado da conversão do primeiro bloco para ser apresentado posteriormente junto com o resultado da conversão do segundo bloco.

Adicionalmente, outra necessidade exibida pelo circuito é a de se armazenar a tensão amostrada " V_{ent} " e a defasar de um período de amostragem, para que ela possa ser convertida pelo segundo bloco.

O resultado da conversão do primeiro bloco é armazenado nos registradores de saída até que o segundo bloco tenha realizado a sua conversão. O sinal de comando para a armazenagem dos dados nos registradores deve atuar pouco antes do resultado de uma nova conversão ser apresentado na saída do codificador do primeiro bloco conversor. Os registradores de saída estão indicados na Fig. 2.11 como "REGS".

Na intenção de se otimizar a conversão e, além disto, suprir a necessidade de se armazenar e defasar a " V_{ent} ", optou-se pelo uso de mais dois circuitos "sample-and-hold" (S/H), além do de entrada. Um deles amostra a diferença entre " V_{ent} e V_{ref-} " e o outro amostra a diferença entre " V_{ref+} " e V_{ref-} ".

As tensões " V_{ref+} " e " V_{ref-} " são as saídas do multiplexador, resultantes da

conversão do primeiro bloco. Estas tensões representam, respectivamente, o limite superior e inferior da faixa em que o segundo bloco de conversão deve atuar.

O primeiro "sample-and-hold" (SH1 na Fig. 2.11) deve amostrar após o primeiro bloco ter apresentado o resultado da primeira conversão e antes do "sample-and-hold" de entrada iniciar uma nova amostragem. Além disso, deve reter o dado amostrado durante a fase de comparação do segundo bloco.

O segundo "sample-and-hold" (SH2) deve amostrar após a apresentação do resultado da conversão do primeiro bloco e antes da fase de auto-zero do segundo bloco.

Um dado válido estará disponível na saída do conversor depois do segundo bloco ter convertido e antes dos registradores de saída atualizarem os seus dados. Durante o intervalo em que o dado está válido, deve ser gerado um sinal tipo habilitador de memória para informar aos dispositivos externos que podem ler os dados. Este sinal deverá ser ativo em nível baixo para se compatibilizar com as memórias comerciais.

O gerador de fases fornece toda a base de tempo para o conversor. Funciona sincronizando as operações internas do circuito. Não se deve esquecer que este bloco também deverá agregar a geração de sinais de sincronização com o exterior.

O elemento mais importante do conversor é, sem dúvida, o comparador. Ele deve ser capaz de decidir se a entrada é maior ou menor que a referência com diferença de $1/4$ do bit menos significativo (LSB), motivo pelo que será tratado na próxima Seção.

2.3 OS COMPARADORES

Diversas aplicações em processamento de sinais necessitam da habilidade de se comparar dois sinais e de se identificar qual é o maior. O comparador realiza esta função emitindo uma saída que depende da magnitude das entradas.

Um grande problema encontrado na comparação de pequenas diferenças de tensão é o do "offset" de entrada. Num conversor de oito bits, trabalhando com sinais de entrada com amplitude em torno de 5V, exige-se um comparador com precisão melhor que 20 mV.

Nos circuitos comparadores integrados MOS, o "offset" de entrada usual dos processos atuais gira em torno de 10 mV, podendo ser maior devido ao "clock feedthrough". Técnicas apuradas de projeto e cuidados na execução do "layout" podem reduzir, mas não eliminar, os efeitos do "offset". O problema agrava-se, neste projeto, pois o processo de difusão utilizado é digital.

Em diversas aplicações, o comparador não opera continuamente. Em outras palavras, em certos casos, ele executa uma comparação entre duas tensões e logo após é zerado. Um caso típico deste tipo de operação é o de comparação de sinais amostrados no tempo.

Quando o comparador opera descontinuamente, é possível se aplicar uma técnica de cancelamento de "offset" chamada auto-zero [18]. Esta técnica funciona particularmente bem com circuitos CMOS devido à alta impedância de entrada dos dispositivos. Durante os instantes em que o comparador sofre "reset",

pode ser feito o auto-zero, melhorando a performance do circuito.

A Fig. 2.12 ilustra o algoritmo de cancelamento do "offset" de entrada. O comparador é considerado como sendo ideal (sem "offset" de entrada) e adiciona-se uma fonte de tensão de polaridade arbitrária ligada à sua entrada, simulando o "offset". Admite-se ainda que o comparador opere com um "clock" de duas fases sem sobreposição.

Durante a primeira fase, o "offset" é medido conforme a Fig. 2.12b e armazenado no capacitor C_{az} . Durante a segunda fase, o capacitor é conectado de tal forma a cancelar os efeitos do "offset", como pode ser visto na Fig. 2.12c.

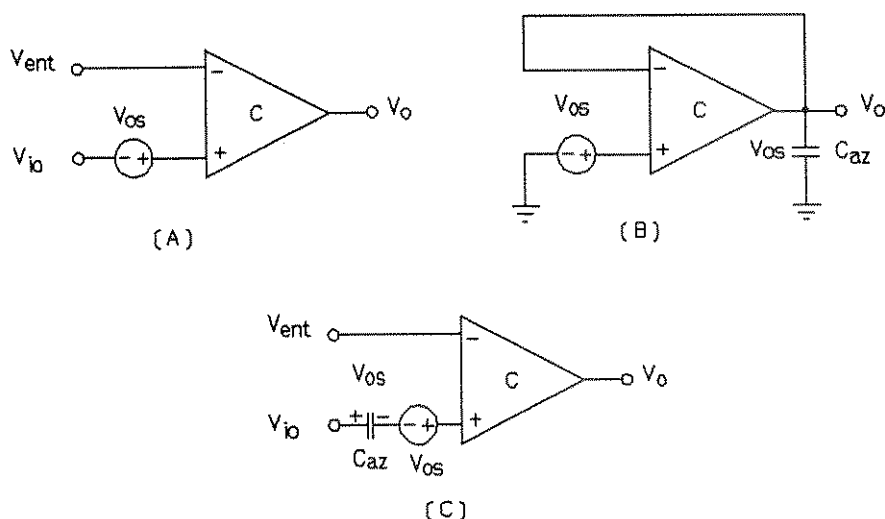


Fig. 2.12 Cancelamento do "Offset" de Entrada de Comparador

A Fig. 2.13 exibe um circuito prático usado para se conseguir a auto-zeragem de um comparador inversor, mostrando as fases ϕ_1 e ϕ_2 .

Embora a técnica de auto-zero pareça a solução perfeita para a eliminação do "offset", ela não remove completamente a sua influência. Um dos motivos é que devido às grandes excursões do sinal de "clock" aplicado nas portas dos transistores, cargas são injetadas ou removidas do canal quando os mesmos se fecham ou se abrem. Estas cargas provocam uma variação na tensão do capacitor C_{az} .

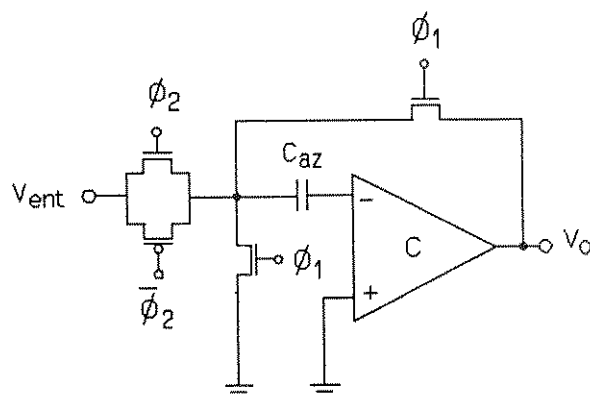


Fig. 2.13 Circuito Prático para Eliminação do "Offset" [18]

Outro motivo é provocado pelo fato do capacitor C_{az} perder parte de sua carga durante a segunda fase em função da redistribuição das cargas entre este capacitor e as capacitâncias parasitas (Fig. 2.14).

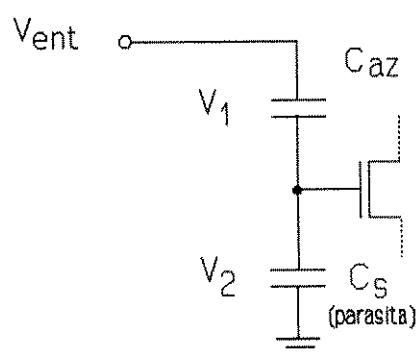


Fig. 2.14 - Redistribuição de Cargas de C_{az}

Finalmente, o terceiro motivo que impede que a técnica de auto-zeragem seja perfeita é caracterizado pela injeção de um ruído do tipo $K.T/C_{az}$, toda vez que um transistor se fecha [18].

Os efeitos de redistribuição de cargas de C_{az} durante a segunda fase podem ser minimizadas fazendo-se C_{az} muito maior que as capacitâncias parasitas ligadas ao nó de entrada do comparador, embora o circuito fique mais lento.

2.3.1 CONFIGURAÇÕES DE COMPARADORES

Há duas configurações de comparadores usadas com maior frequência [19]. A primeira é implementada a partir de uma cascata de inversores, como a indicada na Fig. 2.15, e a segunda fundamenta-se em um estágio diferencial como mostrada na Fig. 2.16.

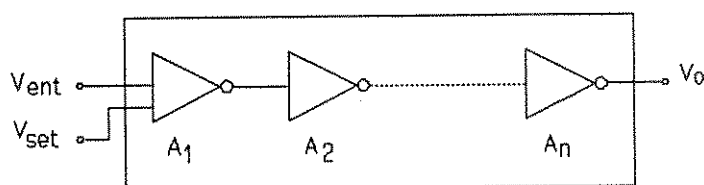


Fig. 2.15 Comparador com Cascata de Inversores

Com a finalidade de melhorar o sinal de saída do comparador, normalmente, qualquer tipo de comparador é seguido de um circuito tipo "latch".

A) COMPARADOR TIPO CASCATA DE INVERSORES

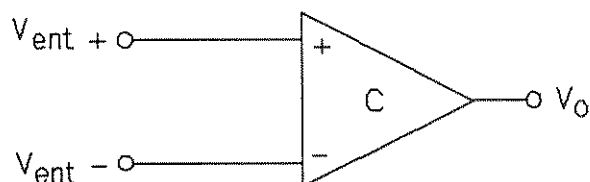


Fig. 2.16 Comparador com Estágio Diferencial

Um sistema como o da Fig. 2.15, implementado em tecnologia CMOS teria como entrada, provavelmente, o estágio inversor "auto-zerado" da Fig. 2.17a. As formas de onda do circuito estão mostradas na Fig. 2.17b.

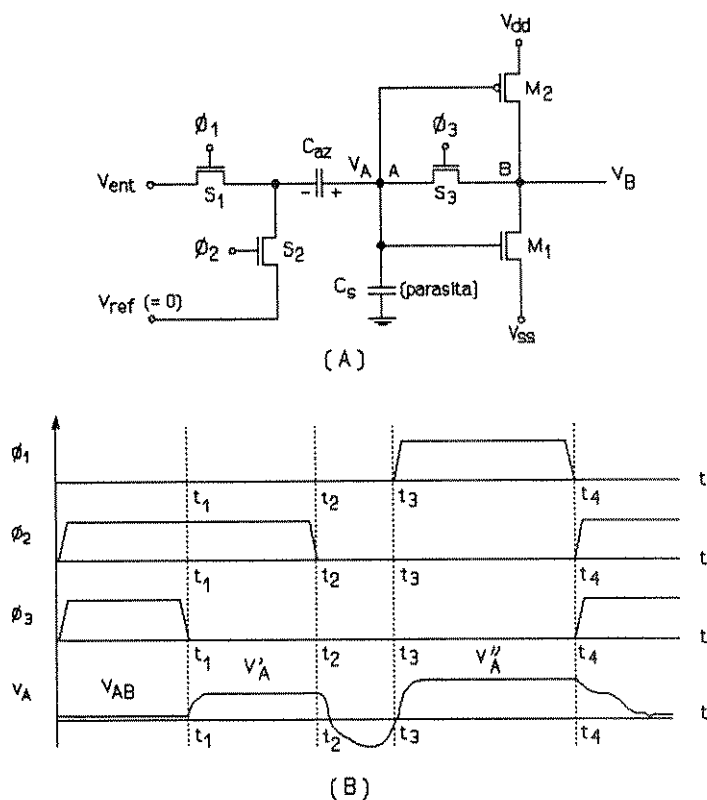


Fig. 2.17 - Comparador Tipo Cascata de Inversores

Em $t = 0$, as chaves S_2 e S_3 estão fechadas. A chave S_2 conecta o terminal do lado esquerdo do capacitor de auto-zero C_{az} à tensão de referência nula, enquanto a chave S_3 põe em curto os nós A e B. Os nós A e B assumem desta forma a tensão dada pela interseção das curvas de transferência do inversor e a reta de 45° (representando a condição de $V_B = V_A$).

A Fig. 2.18 ilustra a situação descrita acima, sendo que a Fig. 2.18a mostra a condição do inversor e a Fig. 2.18b as curvas citadas com $V_{dd} = -V_{ss} = 5V$ e $V_{tn} = -V_{tp} = 1V$.

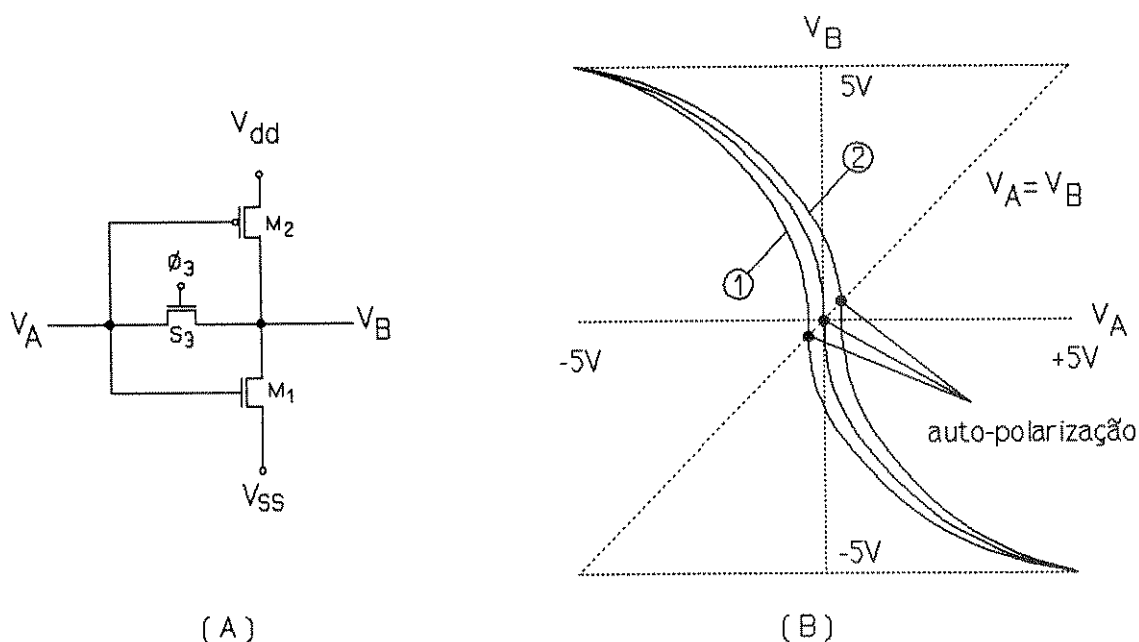


Fig. 2.18 Condições de Polarização para o Inversor de Entrada

A interseção ocorre num ponto onde tanto o transistor M1 quanto M2 estão saturados (meio da faixa linear) e o inversor tem ganho máximo. Esta

interseção é completamente imune às variações das tensões "threshold". Esta situação está ilustrada na Fig. 2.18b curva 1 ($V_{tn} = 0,7V$, $V_{tp} = -1,2V$) e curva 2 ($V_{tn} = 1,2V$, $V_{tp} = -0,7V$). Nos dois casos, o ponto de interseção ocorre próximo ao centro da região linear.

Durante o intervalo de tempo entre $t=0$ e $t=t_1$, o capacitor C_{az} carrega-se até a tensão $V_{az} = V_{AB} - 0 = V_{AB}$, onde V_{AB} é a tensão de interseção (auto-polarização) ilustrada na Fig. 2.18b. Para tensões "threshold" nominais, V_{AB} fica em torno de zero.

Em $t = t_1$, a fase ϕ_3 sofre "reset" desligando a chave S_3 e desconectando os nós A e B. Parte da carga no canal de S_3 é injetada em C_{az} e, através do capacitor de sobreposição porta-fonte da chave, uma carga adicional é injetada em C_{az} devido ao "clock feedthrough" [21].

As dimensões de C_{az} e de S_3 devem ser escolhidas de tal forma que as mudanças em V_A , devido às cargas injetadas, não tirem M1 e M2 de sua região linear de operação. Sejam V_A' e V_B' , as tensões resultantes nos nós A e B, respectivamente.

Em $t = t_2$, a chave S_2 se abre. A não ser pela pequena capacitância parasita C_s , o nó A fica flutuando e, assim sendo, V_A e V_B podem derivar para qualquer valor. Contudo, uma vez que C_{az} comporta-se quase como um circuito aberto no nó A, haverá um pequeno "clock feedthrough" para C_{az} devido ao desligamento de S_2 .

Em $t = t_3$, a fase ϕ_1 é ativada e C_{az} , carregado anteriormente no intervalo de t_1 a t_2 , fica conectado entre a entrada e o nó A. A tensão no nó A fica $V_A'' = V_{ent} + V_A'$ e a diferença de tensão $V_A'' - V_A'$ tem o mesmo sinal e amplitude de V_{ent} .

A última declaração é verdadeira independentemente dos valores de C_s , V_{az} , da magnitude do "clock feedthrough" e de qualquer outro efeito parasita. Assim sendo, monitorando-se a variação $V_B'' - V_B'$ (valor amplificado de V_A'' e de V_A'), pode-se decidir se V_{ent} é maior ou menor que a V_{ref} que neste caso é nula.

B) COMPARADOR COM AMPLIFICADOR DIFERENCIAL

Para comparações de maior exatidão, é preferível o uso de amplificadores diferenciais. Nestes circuitos, o "clock feedthrough", ruídos da alimentação, ruído $1/f$ e outros, tendem a ser cancelados [18].

A Fig. 2.19 apresenta um estágio amplificador diferencial usado como comparador. O algoritmo de cancelamento do "offset" de entrada é o mesmo exposto para o estágio inversor sendo que, neste circuito, o auto-zero é feito nos dois ramos do estágio diferencial.

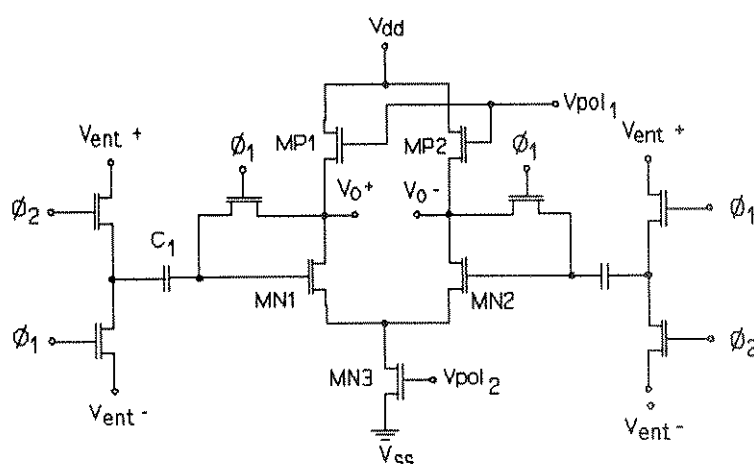


Fig. 2.19 Comparador Diferencial

Apesar das vantagens deste tipo de comparador, o sinal de entrada deve ser limitado em amplitude. Sinais de entrada de grande amplitude levariam o transistor do espelho de corrente (MN3) ao corte e o circuito sairia de sua região normal de operação.

2.3.2 O COMPARADOR PROPOSTO NO PROJETO

Na Fig. 2.20 é mostrado o diagrama em blocos do comparador usado neste projeto. Trata-se de um comparador do tipo cascata de inversores constituído de um estágio comparador de entrada, um estágio amplificador e um estágio "latch".

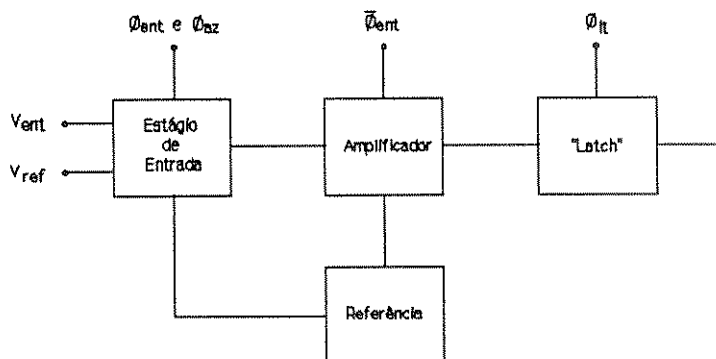


Fig. 2.20 Esquema do Comparador Usado no Projeto

O estágio comparador de entrada pode ser visto na Fig. 2.21. O seu funcionamento é bastante semelhante ao do estágio de entrada descrito na Seção anterior. As diferenças decorrem do fato deste estágio ser constituído por dois inversores casados em série e dos transistores tipo p estarem ligados a uma

referência fixa. Os inversores são amplificadores com carga p ativa que possuem um ganho menor que o do inversor CMOS de mesmas dimensões.

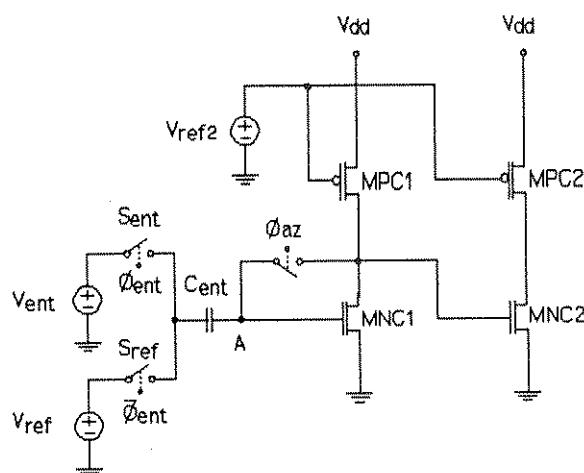


Fig. 2.21 Estágio Comparador de Entrada

Conseguiu-se, com este inversor, reduzir a capacitância (parasita) de entrada do estágio. Este tipo de inversor permite a diminuição da capacitância de auto-zero (Cent) que, conforme foi visto anteriormente, deve ser bem maior que a capacitância parasita no nó A.

O ganho menor apresentado por um inversor é compensado colocando-se dois inversores em série. Os dois estágios inversores devem ser casados de tal forma que as variações na polarização devido à temperatura, variações de alimentação e tolerâncias na fabricação sejam as mesmas em ambos os estágios.

Com as considerações acima atendidas, pode-se garantir que quando for

feito o auto-zero no primeiro estágio, tanto este quanto o segundo estágio estarão polarizados no mesmo ponto e, conforme visto anteriormente, na região linear de operação e com um ganho próximo do máximo.

A referência aplicada nas portas dos transistores P deve ser próxima de $V_{dd}/2$ para que, projetados os estágios adequadamente, o ponto de auto polarização seja próximo de $V_{dd}/2$. Este procedimento garantirá que o sinal tenha excursões iguais para cima e para baixo do ponto de polarização.

O estágio amplificador, que pode ser visto na Fig. 2.22, serve para aumentar a excursão do sinal de saída do estágio comparador. Ele entrega ao "latch" um sinal "forte", diminuindo o tempo necessário para que este se ajuste.

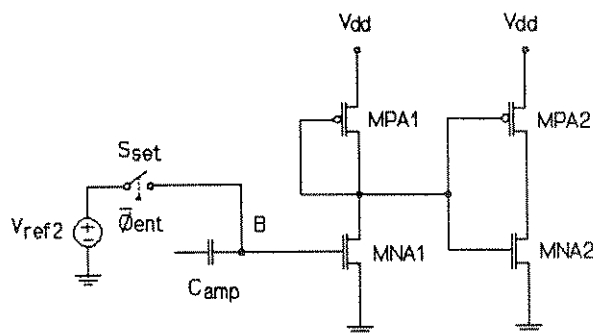


Fig. 2.22 - Estágio Amplificador

Este estágio está acoplado ao estágio comparador através de um capacitor C_{amp} que remove a componente dc do sinal. A importância de se eliminar essa componente do sinal reside no fato de que, em uma cascata de inversores com alto ganho, um pequeno desvio no ponto de polarização pode levar um dos estágios

a operar fora de sua região linear, com consequente redução de ganho.

Este estágio é polarizado em sua região linear de trabalho por um método diferente do usado para se polarizar o estágio de entrada. Uma tensão de referência de valor $V_{dd}/2$ é aplicada ao nó B durante o tempo em que o estágio de entrada sofre o auto-zero. Os dois inversores que formam este estágio são projetados de forma a terem suas polarizações próximas a $V_{dd}/2$, permitindo uma excursão simétrica do sinal. Enquanto a referência estiver ligada ao nó B, os dois inversores estarão no seus pontos de maior ganho, tornando o estágio bastante sensível.

O primeiro inversor está ligado numa configuração de amplificador com transistor P ligado como diodo de carga. Esta configuração apresenta um ganho baixo e permite que se diminua a capacitância de entrada do estágio.

A redução de capacitância ocorre por se ter apenas uma porta conectada na entrada, adicionado ao fato do "Efeito Miller" ter sido atenuado devido ao ganho baixo. O circuito fica, assim, mais rápido e se evita a divisão de tensão entre C_{amp} e a capacitância parasita de entrada.

O segundo inversor é do tipo CMOS com alto ganho, compensando o ganho baixo do primeiro inversor.

O estágio "latch" tem as funções de reter o valor da comparação anterior enquanto uma nova comparação é executada e aumentar a excursão do sinal de saída, reduzindo os tempos de subida e descida do sinal. Ele serve como interface para os circuitos digitais que vão processar a saída do comparador. O "latch" usado neste comparador pode ser visto na Fig. 2.23.

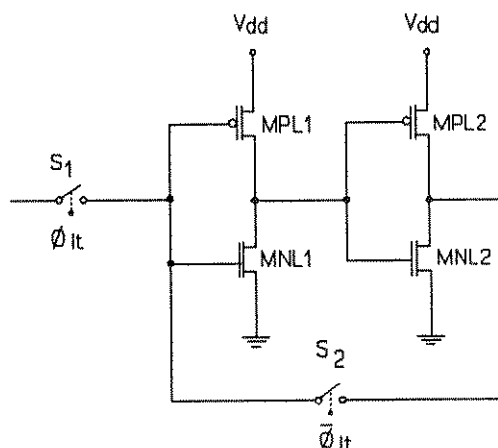


Fig. 2.23 - O Estágio "Latch"

O funcionamento do "latch" é bastante simples. Quando a chave S_1 está fechada e S_2 está aberta, o sinal de saída do estágio amplificador é aplicado na entrada dos dois inversores em série. Quando se abre S_1 e se fecha S_2 , uma realimentação positiva é realizada e o sinal é memorizado.

Deve-se levar em conta que o sinal na entrada do "latch" está quase sempre saturado em V_{dd} ou GND e os inversores CMOS do "latch" têm alto ganho. Com isto, o sinal de saída do "latch" é um sinal digital potente e de amplitude bem definida.

A Fig. 2.24 mostra o circuito completo do comparador juntamente com as fases de "clock" para ilustrar o funcionamento do conjunto.

Em $t = t_0$ (Fig. 2.17), as fases ϕ_{ent} e ϕ_{lt} estão zeradas e ϕ_{az} está ativada. Assim, as chaves S_{ent} e S_1 estão abertas e S_{ref} , S_{set} , S_{az} e S_2 estão fechadas.

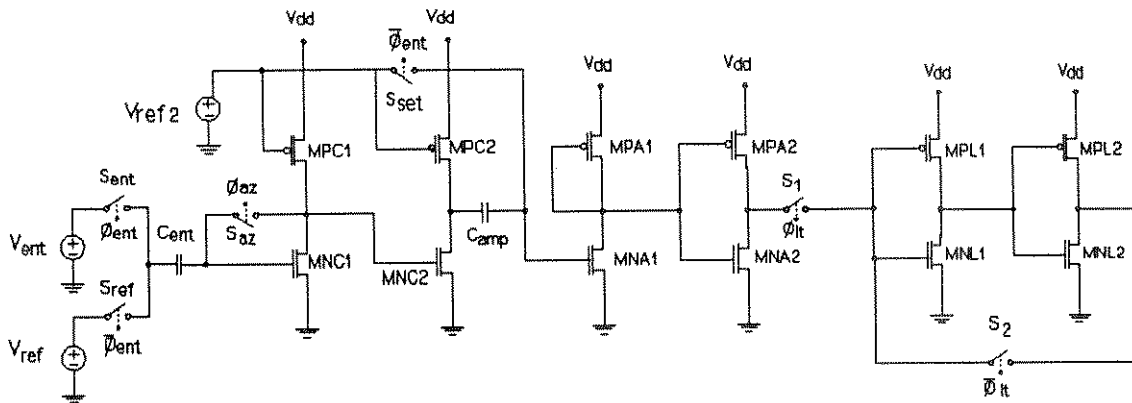


Fig. 2.24 Circuito Completo do Comparador

O estágio de entrada está fazendo o auto-zero e carregando em C_{ent} a diferença entre a tensão de auto-polarização e a tensão de referência (V_{ref}). O estágio amplificador está sendo polarizado no seu ponto de maior ganho pela tensão $V_{ref2} = V_{dd}/2$. O "latch" está retendo um dado qualquer não válido.

Em $t = t_1$, a fase ϕ_{az} é desativada e a chave S_{az} é aberta. Assim, o circuito está pronto para a comparação com V_{ent} .

Em $t = t_2$, a ϕ_{ent} é ativada, S_{ent} é fechada e S_{ref} e S_{set} são abertas. O circuito de entrada faz a comparação e entrega para o estágio amplificador o resultado da comparação. O amplificador, agora desconectado de V_{ref2} , resolve rapidamente para cima ou para baixo dependendo do resultado da comparação.

O intervalo de t_2 a t_3 corresponde ao tempo necessário para que o sinal se estabilize.

Em $t = t_3$, ϕ_{lt} é levada ao nível lógico 1, S_1 é fechada e S_2 é aberta. O "latch" atualiza o dado de saída que agora é um dado válido.

Após um intervalo de tempo necessário para o estágio "latch" responder, em $t = t_4$, ϕ_{lt} é zerada, S_1 é aberta e S_2 fechada. Desta maneira, o dado é memorizado e não varia mais até a próxima fase de comparação.

Em $t = t_5$, uma nova fase de comparação é iniciada.

Pode-se notar no circuito da Fig. 2.24 que cada estágio é formado por dois inversores tornando-se, portanto, estágios não inversores. Com isto, se

$$V_{ent} > V_{ref} \rightarrow saída = 1$$

então,

$$V_{ent} < V_{ref} \rightarrow saída = 0.$$

2.4 PROJETO DOS BLOCOS DO CONVERSOR

2.4.1 O COMPARADOR

A) OS TRANSISTORES-CHAVE

As chaves do comparador, formadas por transistores, serão todas

de tamanho mínimo para reduzir os efeitos da injeção de carga do canal na capacitância de auto-zero alterando a polarização dos inversores. Assim,

$$W_{ch} = 3\mu$$

e

$$L_{ch} = 2\mu,$$

que são as dimensões mínimas permitidas pelo processo ES2 no PMU CMOS 5.

B) O ESTÁGIO DE ENTRADA

O estágio de entrada do comparador já foi mostrado na Fig. 2.21. Para se reduzir os efeitos de variação do comprimento do canal (durante a fabricação) na polarização do estágio de entrada, é recomendável que se trabalhe com comprimentos de canal não menores que 6μ . Levando-se em conta esta limitação, foi adotado um transistor MNC1 de dimensões:

$$W_{MNC1} = 6\mu$$

e

$$L_{MNC1} = 6\mu.$$

Através de simulação, ajustou-se as dimensões de MPC1 de forma a se obter a polarização próxima de $VDD/2 = 2,5V$. Empregou-se, para este fim, o circuito mostrado na Fig. 2.25.

Como na região linear de operação do inversor, os dois transistores estão saturados, pode-se escrever que:

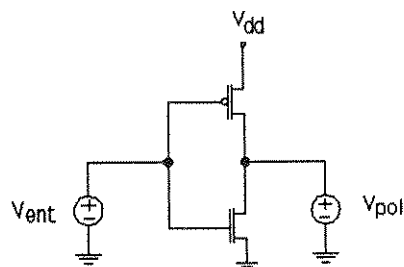


Fig. 2.25 Circuito de Ajuste da Polarização

$$I_{dmn} = k_n \cdot \frac{W_{mn}}{L_{mn}} \quad (2-7)$$

e

$$I_{dmp} = k_p \cdot \frac{W_{mp}}{L_{mp}} \quad (2-8)$$

Obtendo-se as correntes I_{dmn} e I_{dmp} para os respectivos valores de W e L , pode-se obter k_n e k_p . No esquema da Fig. 2.25, para que a tensão na saída seja $V_{dd}/2$, a corrente na fonte de valor $V_{dd}/2$ deve ser nula. Logo, $I_{dmp} = I_{dmn}$, onde:

$$\frac{I_{dmn}}{I_{dmp}} = \frac{k_n}{k_p} \cdot \frac{W_{mn}}{W_{mp}} \cdot \frac{L_{mp}}{L_{mn}} = 1 \quad (2-9)$$

Como $L_{mn} = L_{mp} = 6\mu$, tem-se:

$$W_{mp} = W_{mn} \cdot \frac{k_n}{k_p} \approx W_{mn} (2.5) \quad (2-10)$$

Assim, obtém-se como valores práticos:

$$W_{MPC1} = 15\mu$$

$$W_{MPC1} = 6\mu$$

Como os dois inversores são casados,

$$W_{MNC2} = 6\mu$$

$$L_{MNC2} = 6\mu$$

$$W_{MPC2} = 15\mu$$

$$L_{MPC2} = 6\mu.$$

É aconselhável que a capacitância C_{ent} seja pelo menos dez vezes maior que as capacitâncias parasitas da entrada, formadas pelas capacitâncias de porta do MNC1. De acordo com as simulações feitas para a determinação da polarização do inversor, as capacitâncias de gate de MNC1 valem:

$$C_{gsov1} = 9 \times 10^{-16} F$$

$$C_{gdov1} = 9 \times 10^{-16} F$$

$$C_{gbov1} = 0$$

$$C_{gs} = 1,97 \times 10^{-16} F$$

$$C_{gd} = 0$$

$$C_{gb} = 0$$

Logo,

$$C_s = C_{gsov1} + C_{gdov1} + C_{gbov1} + C_{gd} + C_{gb}$$

ou,

$$C_s = 2,15 \times 10^{-14} = 0,02 \text{ pF}.$$

Assim,

$$C_{ent} = 10.C_s$$

ou,

$$C_{ent} = 0,2 \text{ pF.}$$

C) O ESTÁGIO AMPLIFICADOR

O dimensionamento deste estágio, que pode ser visto na Fig. 2.22, leva em consideração o comprimento de canal mínimo igual a 6μ . Procurou-se trabalhar com as dimensões mínimas. O segundo estágio, no entanto, necessitou ser aumentado a fim de alimentar corretamente o "latch", reduzindo os tempos de atraso sem aumentar em muito o consumo. Partiu-se das dimensões iguais às do estágio de entrada, visando obter as mesmas tensões de polarização. Através de ajustes por simulação, chegou-se a:

$$W_{MNA1} = 6\mu$$

$$L_{MNA1} = 6\mu$$

$$W_{MPA1} = 15\mu$$

$$L_{MPA1} = 6\mu$$

$$W_{MNA2} = 12\mu$$

$$L_{MNA2} = 6\mu$$

$$W_{MPA2} = 30\mu$$

$$L_{MPA2} = 6\mu$$

A capacitância de entrada do estágio amplificador é aproximadamente igual à do estágio de entrada, pois os transistores de entrada são de mesmas dimensões. Logo:

$$C_{amp} = C_{ent}$$

$$C_{amp} = 0,02\text{pF}$$

D) O ESTÁGIO "LATCH"

Projetou-se este estágio, mostrado na Fig. 2.23, levando-se em consideração o comprimento mínimo de canal igual a 6μ . Objetivando reduzir o consumo, trabalhou-se com pequenas larguras de canal. Como o sinal na entrada deste estágio tem uma grande excursão, não é necessário que o ponto de decisão do circuito seja em $V_{dd}/2$. Assim:

$$W_{MNI1} = 6\mu$$

$$L_{MNI1} = 6\mu$$

$$W_{MPI1} = 6\mu$$

$$L_{MPI1} = 6\mu$$

$$W_{MNI2} = 6\mu$$

$$L_{MNI2} = 6\mu$$

$$W_{MPI2} = 6\mu$$

$$L_{MPI2} = 6\mu$$

A Fig. 2.24, anteriormente apresentada, mostra o diagrama esquemático final do comparador submetido à simulações elétricas nos casos: típico, pior caso de velocidade, pior caso de potência, pior "0" e pior "1".

A discussão já realizada sobre o comparador mostra que para se obter as frequências das fases e consequentemente a do comparador, dois tempos merecem uma

análise criteriosa na simulação:

- o tempo que o capacitor de entrada necessita para se carregar durante a fase de auto-zero;
- o tempo necessário para que o sinal de entrada alcance a entrada do "latch" durante a fase de entrada.

Como o "latch" é extremamente rápido e a fase de "latch" ocorre durante a fase de auto-zero, o tempo de propagação do sinal de entrada por este circuito será bem menor que o necessário para se carregar o capacitor de entrada. Por este motivo, não é necessária a determinação da temporização do "latch".

A Tab. 2.1 mostra o tempo de carregamento do capacitor de entrada considerando em todos os casos, três valores de tensão "threshold" para que a simulação elétrica possa conduzir a resultados mais efetivos.

A Tab. 2.2 mostra o tempo que o sinal de entrada leva para se propagar pelo estágio de entrada, pelo amplificador e indo até a entrada do circuito "latch" do comparador. Os valores indicados refletem o pior caso de propagação, quer seja na transição de nível baixo para alto ou de alto para baixo.

Observou-se, durante as simulações do comparador, que ele exibiu uma ligeira instabilidade quando eram aplicadas pequenas diferenças de tensão nas entradas (da ordem de 4mV) em algumas condições (temperatura, alimentação, tensão "threshold" e tipo de transistor).

Acredita-se que os erros no estágio de entrada sejam devidos a problemas

Blocos CMOS de Alta Performance para Aplicações em VLSI

CASOS	TENSÃO "THRESHOLD"	TEMPO MÁX. CARREGAMENTO (nS)
Típico	4.5 V	20.0
	2.5 V	12.0
	0.5 V	10.0
Pior Caso de Velocidade	4 V	25.0
	2.25 V	20.0
	0.5 V	18.0
Pior Caso de Potência	5.0 V	10.0
	2.75 V	8.0
	0.5 V	7.0
Pior "0"	4.0 V	40.0
	2.25 V	30.0
	0.5 V	1.0
Pior "1"	4.0 V	20.0
	2.25 V	18.0
	0.5 V	10.0

Tabela 2.1 Tempo de Carregamento do Capacitor de Entrada

no simulador que não deve ser preciso para circuitos chaveados e transferência de cargas com pequenas diferenças de tensão. Chegou-se a esta conclusão após várias simulações e tentativas de se ajustar o circuito de entrada.

Blocos CMOS de Alta Performance para Aplicações em VLSI

CASOS	TENSÃO "THRESHOLD"	TEMPO MÁX.PROPAGAÇÃO (nS)
Típico	4.5 V	12.0
	2.5 V	32.0
	0.5 V	11.0
Pior Caso de Velocidade	4 V	31.0
	2.25 V	68.0
	0.5 V	43.0
Pior Caso de Potência	5.0 V	10.0
	2.75 V	7.5
	0.5 V	8.6
Pior "0"	4.0 V	10.0
	2.25 V	12.0
	0.5 V	19.0
Pior "1"	4.0 V	58.0
	2.25 V	53.0
	0.5 V	6.0

Tabela 2.2 Propagação do Sinal de Entrada até o "Latch"

Tentou-se aumentar o capacitor C_{ent} , as chaves e os transistores (em conjunto e separadamente). Observou-se que os resultados não condiziam com as expectativas e que variações nos tempos de simulação e no passo de simulação

causavam variações significativas nas respostas do circuito.

Como o circuito se constitui numa estrutura de teste, decidiu-se difundir o circuito e realizar medições no protótipo a fim de se verificar o funcionamento do circuito e a validade das respostas obtidas nas simulações.

Uma das considerações mais importantes do projeto pode ser conseguida mediante a análise da Tab. 2.1. O pior tempo de carregamento do capacitor de entrada ocorre no caso de pior "0", situação em que os transistores N são lentos, com valor igual a 40,0 nS e para uma tensão "threshold" de 4 V. Como este tempo está amarrado à fase de auto-zero dos comparadores, conclui-se que, na pior situação, esta fase deverá ficar ativa durante um tempo mínimo de 40 nS.

Outro tempo crítico pode ser obtido na Tab. 2.2, para o pior caso de velocidade em que os transistores são lentos, correspondendo à propagação do sinal de entrada através dos comparadores até atingir o estágio "latch", com valor igual a 68,0 nS. Este valor corresponde ao tempo em que a fase de entrada deverá ficar ativa.

Os dois tempos indicados acima limitam a operação do conversor. Quando os comparadores estão sendo auto-zerados, o restante da parte digital está processando a última conversão e levando-a à saída. O tempo de auto-zero é muito maior que o necessário à resposta dos blocos digitais, motivo pelo qual não se preocupou com os tempos de propagação destes últimos.

Adicionalmente, quando o sinal de entrada está se propagando pelo comparador, nenhuma outra atividade deverá estar ocorrendo, o que faz com que este tempo também seja limitante na frequência de conversão.

Com as considerações acima e, com base na Fig. 2.31, conclui-se que a máxima frequência de conversão em pior caso é de :

$$\text{Máxima Frequência} = \frac{1}{(40+68)} nS \approx 10 \text{ MHz}$$

Uma análise semelhante à realizada acima, apenas para o caso típico, conduziria a uma frequência de operação em torno de 20 MHz. Em ambos os casos, os resultados são altamente expressivos e encorajadores.

2.4.2 OS RESISTORES DE REFERÊNCIA

Levando-se em consideração o consumo e o correto carregamento dos capacitores de auto-zero, adotou-se como aceitável uma corrente quiescente pelos resistores igual a 400 μA . Logo:

$$R_{tot} = 16xR = 5V/400\mu A$$

ou,

$$R = 3.125/4 \text{ ou } R = 750 \text{ Ohms}$$

2.4.3 O CODIFICADOR DE 15 ENTRADAS E 4 SAÍDAS

A Tab. 2.3 apresenta uma relação entre as entradas de E1 a E15 e as

Blocos CMOS de Alta Performance para Aplicações em VLSI

E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12	E13	E14	E15	S1	S2	S3	S4
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	1
0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	1	0	0
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	1	0	1
0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	1	1	0
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	1	1	1
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0
0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	1
0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	1	0
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Tabela 2.3 - Tabela Lógica do Codificador

saídas S1 a S4 que devem ser executadas pelo codificador do projeto. Assim, com base nesta tabela, podem-se deduzir as equações de S1, S2, S3 e S4, apresentadas na sequência:

$$S1 = E8 \quad (2-11)$$

$$S2 = E4 + E12 \cdot \overline{E8} \quad (2-12)$$

$$S3 = E2 + \overline{E4} \cdot E6 + \overline{E8} \cdot E10 + \overline{E12} \cdot E14 \quad (2-13)$$

$$S4 = E1 + \overline{E2} \cdot E3 + \overline{E4} \cdot E5 + \overline{E6} \cdot E7 + \overline{E8} \cdot E9 + \overline{E10} \cdot E11 + \overline{E12} \cdot E13 + \overline{E14} \cdot E15 \quad (2-14)$$

As considerações feitas para a obtenção das equações acima baseiam-se no fato de somente os estados indicados serem possíveis de ocorrência. A combinação acima não ocorre em caso de defeito em um dos comparadores ou na inicialização do conversor (quando os dados não são válidos).

Trabalhando-se as equações acima de forma a possibilitar a sua implementação apenas com portas lógicas de duas entradas, obtém-se:

$$S1 = E8 \quad (2-15)$$

$$S2 = \overline{\overline{E4} \cdot \overline{E12} \cdot \overline{E8}} \quad (2-16)$$

$$S3 = \overline{\overline{E2} \cdot \overline{E4} \cdot E6 + \overline{E8} \cdot E10 \cdot \overline{E12} \cdot E14} \quad (2-17)$$

$$\overline{S4} = \overline{\overline{E1} \cdot \overline{E2} \cdot \overline{E3} + \overline{E4} \cdot \overline{E5} \cdot \overline{E6} \cdot \overline{E7} + \overline{E8} \cdot \overline{E9} \cdot \overline{E10} \cdot \overline{E11} + \overline{E12} \cdot \overline{E13} \cdot \overline{E14} \cdot \overline{E15}}$$

(2-18)

Os blocos lógicos do codificador foram projetados com transistores de pequenas dimensões. O comprimento de canal é o mínimo (2μ) e a largura é pouco maior que a mínima (4μ).

A Fig. 2.26 mostra o diagrama esquemático do codificador utilizado neste projeto e que foi submetido a simulação elétrica nos piores casos, com resultados apresentados na Tab. 2.4 para o tempo de propagação de um sinal de entrada para a saída, através do caminho mais crítico.

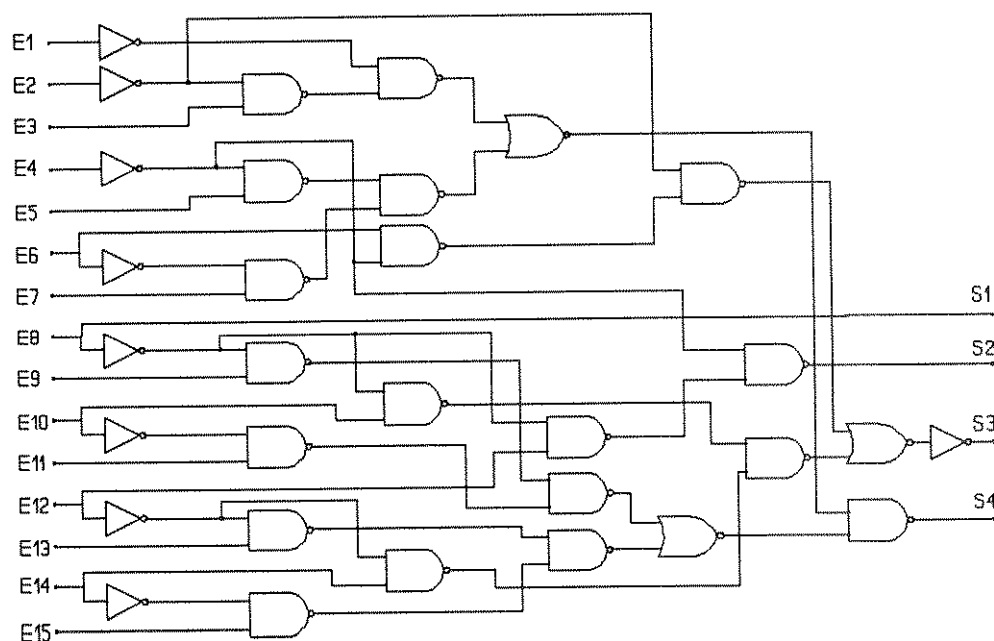


Fig. 2.26 O Codificador de 15 Entradas e 4 Saídas

CASOS	TEMPO DE PROPAGAÇÃO (nS)	
	Baixo para Alto	Alto para Baixo
Típico	1.6	2.0
Pior Caso de Velocidade	3.0	3.8
Pior Caso de Potência	0.9	1.2
Pior "0"	2.1	2.6
Pior "1"	2.2	2.3

Tabela 2.4 Temporização do Codificador

2.4.4 O GERADOR DAS FASES DE "CLOCK" E SINAIS DE CONTROLE

O "clock" de entrada deve permitir que, a partir dele, possa se gerar as fases ϕ_{ent} , $\overline{\phi}_{ent}$, ϕ_{az} , $\overline{\phi}_{az}$, ϕ_{lt} e $\overline{\phi}_{lt}$. Assim, o gerador de fases pode ser implementado de diversas formas. A solução mais simples encontrada é a da Fig. 2.27a. A Fig. 2.27b mostra as formas de onda do "clock" de entrada e das fases que devem ser geradas pelo circuito.

A Fig. 2.28 ilustra a geração de cada fase de "clock" separadamente. Um flip-flop tipo D é usado para dividir a frequência de entrada por dois, resultando num pulso quadrado (sinal Q).

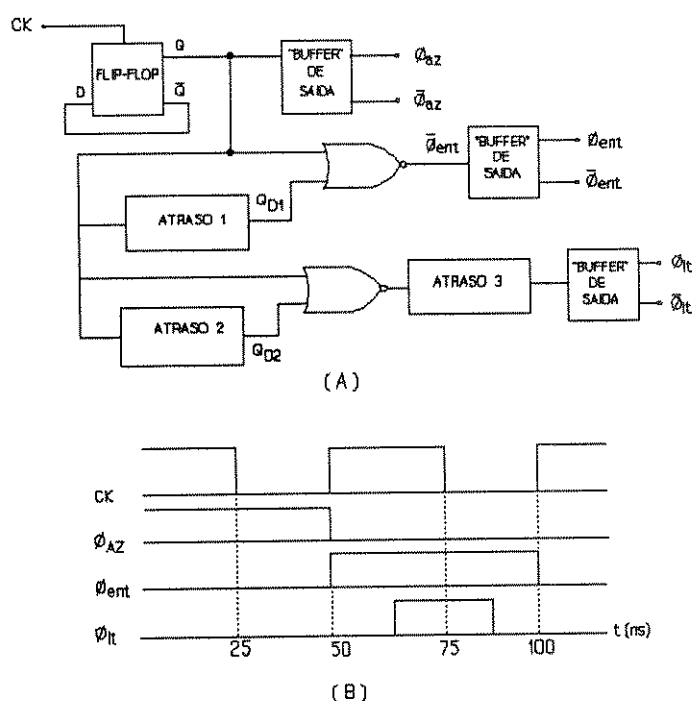


Fig. 2.27 O Gerador de Fases

As saídas do flip-flop (Q e $\bar{Q}$) são aplicadas a blocos lógicos juntamente com estas mesmas saídas atrasadas no tempo (atrasos D1, D2 e D3). Através de lógica combinacional, gera-se todas as fases necessárias.

Uma vez que o circuito deve alimentar dois conjuntos de quinze comparadores, é necessário que as fases passem por um "buffer". Deve-se, então, estimar a carga da cada fase. Para isto, é necessário que se determine a capacitância de um gate de chave, além de se determinar o número de gates alimentados por cada fase do circuito.

A capacitância de gate de uma chave pode ser determinada usando-se

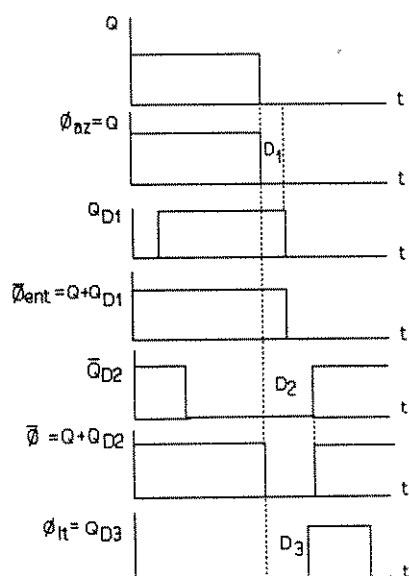


Fig. 2.28 Ilustração do Processo de Geração de Fases

o esquema da Fig. 2.29. Simulando-se uma chave segundo este esquema, obtiveram-se os valores de capacitância da Tab. 2.5.

Somando-se as capacitâncias encontradas por simulação, obtiveram-se os valores de capacitância de cada transistor da chave, ou seja, NMOS = $3,84 \times 10^{-15} \text{F}$ e PMOS = $4,02 \times 10^{-15} \text{F}$.

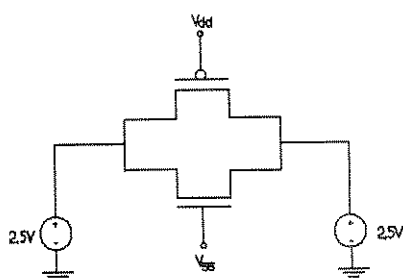


Fig. 2.29 Determinação de Cch

	NMOS	PMOS
Cgsovi =	$4.5 \times 10^{-16} \text{F}$	$6.3 \times 10^{-16} \text{F}$
Cgdovi =	$4.5 \times 10^{-16} \text{F}$	$6.3 \times 10^{-16} \text{F}$
Cgbovi =	0	0
Cgs =	$2.94 \times 10^{-15} \text{F}$	$2.76 \times 10^{-15} \text{F}$
Cgd =	0	0
Cgb =	0	0

Tabela 2.5 - Capacitâncias de Gate das Chaves

Para facilidade dos cálculos, adotou-se o mesmo valor para os dois transistores e definiu-se $C_{ch} = 4 \times 10^{-15} \text{ F}$ (capacitância de porta da chave).

Cada fase de "clock" vai ligada a um certo número de portas de chaves. Na Tab. 2.6 tem-se o levantamento das cargas das fases de "clock".

Empregando-se uma carga de 0,4 pF, igual para todas as fases, ajustou-se, através de simulações, o gerador de fases. O flip-flop foi feito com as dimensões mínimas da tecnologia. O inversor da saída $\overline{Q}$ possui uma carga maior que os demais inversores do flip-flop e, por isto, tem dimensões maiores que as mínimas.

O sinal de "habilitação da memória" deve ser gerado sempre que o dado na saída do "latch" estiver pronto para ser lido por um dispositivo externo. Como o conversor deve ser ligado a memórias comerciais, o sinal deve ser ativo em nível baixo.

Fase	No. de gates x Cch	Carga
ϕ_{ent}	90 x 4fF	360fF ou 0.4pF
$\overline{\phi}_{ent}$	90 x 4fF	360fF ou 0.4pF
ϕ_{zz}	30 x 4fF	120fF ou 0.15pF
$\overline{\phi}_{zz}$	30 x 4fF	120fF ou 0.15pF
ϕ_{lt}	60 x 4fF	240fF ou 0.25pF
$\overline{\phi}_{lt}$	60 x 4fF	240fF ou 0.25pF

Tab. 2.6 Carga das Fases de Clock

O sinal $\overline{\phi}_{lt}$ está em nível alto durante o tempo em que o sinal no "latch" está variando e em nível baixo quando o sinal no "latch" está estável. Para simplificar o circuito, usou-se ϕ_{lt} como "habilitação de memória".

O diagrama esquemático final do gerador de fases está na Fig.2. 30. Deve-se observar que no "buffer" de saída não se usou o fator de ampliação "e = 2.718281828....." (base neperiana) entre o segundo estágio e o primeiro, visto que, a carga do primeiro estágio é maior que a do segundo.

Levando-se em consideração a proposta tratada na Seção 2.2.2, além das fases discutidas acima, é necessária a geração dos seguintes sinais de controle:

- controle dos registradores de saída " ϕ_{reg} ";
- saída de habilitação de memória " ϕ_{em} ";

- controle do "sample-and-hold" de entrada " ϕ_{sh0} ";
- controle do "sample-and-hold" de V_{ent} e de V_{ref} " ϕ_{sh1} ";
- controle do "sample-and-hold" de V_{ref+} e de V_{ref-} " ϕ_{sh2} ".

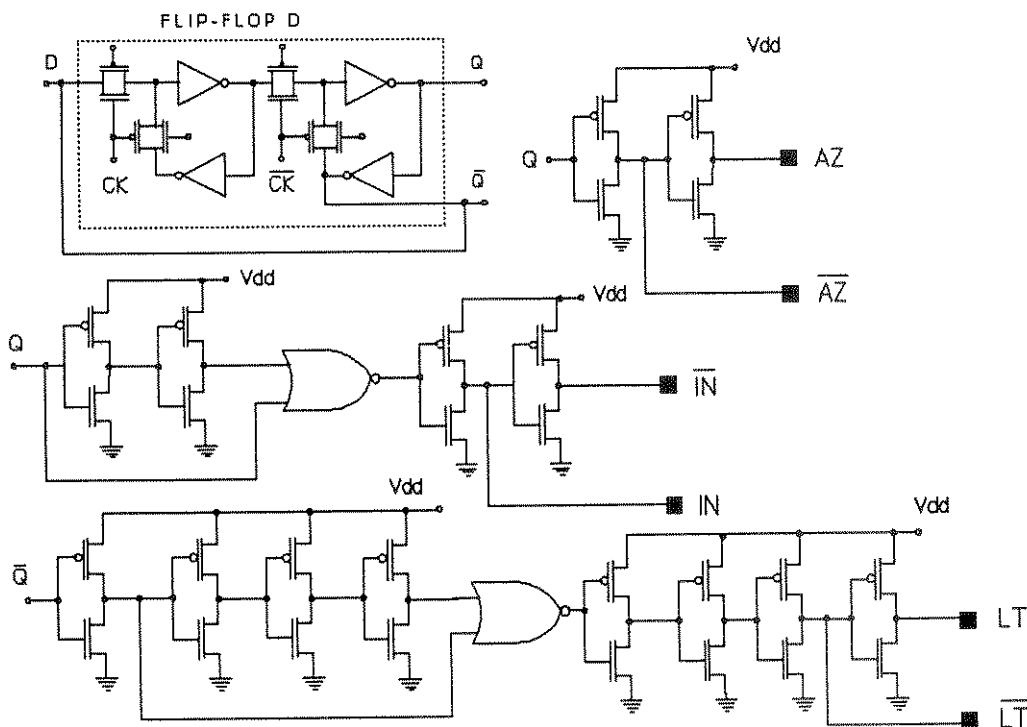


Fig. 2.30 Diagrama do Gerador de Fases

Os sinais acima devem ser sincronizados com os outros já gerados no conversor. Assim, preferiu-se gerá-los a partir dos sinais das fases já implementados. A Fig. 2.31 mostra todos os sinais e as suas temporizações.

Para se chegar ao circuito de geração dos sinais de controle, utilizou-se o circuito mostrado na Fig. 2.32a que gera um sinal genérico "c" a partir dos sinais "a"

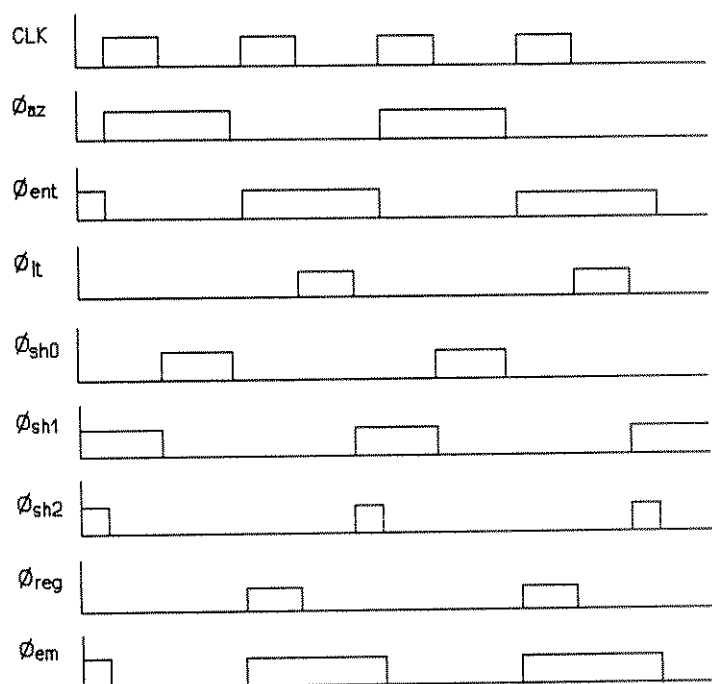


Fig. 2.31 Sinais de Controle do Conversor A/D

e "b" (Fig. 2.32b) da seguinte maneira:

- o sinal "c" vai para "1" com a descida do sinal "b";
- o sinal "c" vai para "0" com a subida de "a".

As equações que seguem, mostram a representação booleana da função executada pelo circuito da Fig. 2.32, onde Y é uma variável auxiliar:

$$c = Y.\overline{b}$$

e,

$$\bar{Y} = a + \bar{b} \cdot \bar{Y}$$

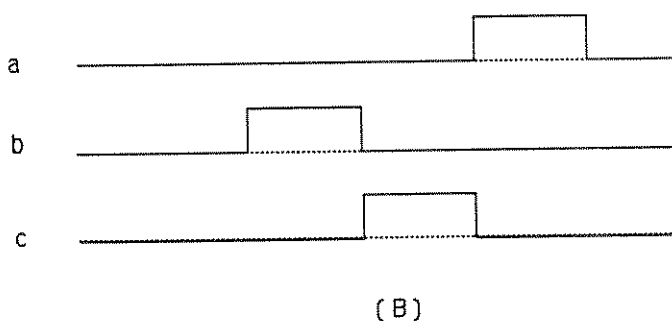
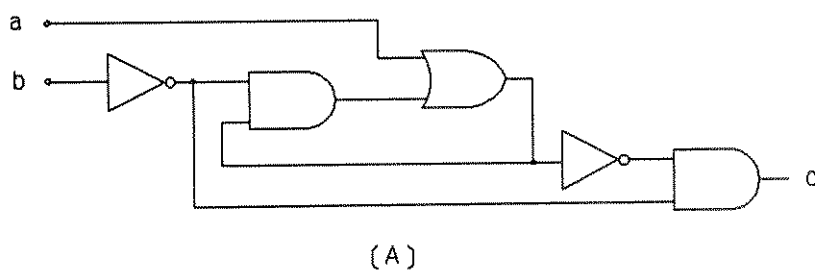


Fig. 2.32 Circuito Lógico de Geração do Sinal "c"

De acordo com o funcionamento do conversor e, analisando-se os sinais da Fig. 2.31, pode-se obter ϕ_{sh1} , ϕ_{sh2} e ϕ_{reg} utilizando-se o circuito da Fig. 2.32a. O sinal ϕ_{sh2} vai para "1" na descida de ϕ_{lt} e para "0" na subida de ϕ_{az} . Substituindo-se nas equações acima, "c" por ϕ_{sh2} , "b" por ϕ_{lt} e "a" por ϕ_{az} , obtém-se:

$$\phi_{sh2} = Y \cdot \overline{\phi_{lt}}$$

$$\bar{Y} = \phi_{az} + \overline{\phi_{lt}} \cdot \bar{Y}$$

De forma análoga ao caso anterior, o sinal ϕ_{sh1} vai para "1" com a descida do sinal ϕ_{lt} e para "0" com a subida de ϕ_{sh0} . Assim,

$$\phi_{sh1} = Y \cdot \overline{\phi_{lt}}$$

$$\overline{Y} = \phi_{sh0} + \overline{\phi_{lt}} \cdot \overline{Y}$$

O sinal ϕ_{reg} vai para "1" na subida de ϕ_{ent} e para "0" na subida de ϕ_{lt} . Como consequência,

$$\phi_{reg} = Y \cdot \phi_{ent}$$

$$\overline{Y} = \phi_{lt} + \phi_{ent} \cdot \overline{Y}$$

O sinal ϕ_{sh0} deve ser ativo durante a fase de auto-zero e deve iniciar após o término de ϕ_{sh1} . Como este último sinal foi gerado a partir do próprio ϕ_{sh0} , o intertravamento é garantido entre eles.

De forma a se permitir a flexibilidade da duração de ϕ_{sh0} e de ϕ_{sh1} , optou-se por gerar ϕ_{sh1} a partir do sinal de "clock" externo CLK. Pode-se observar na Fig. 2.32 que ϕ_{sh0} é ativo quando ϕ_{az} é "1" e CLK é "0". Logo,

$$\phi_{sh0} = \phi_{az} \cdot \overline{CLK}$$

ou ainda,

$$\phi_{sh0} = \overline{CLK + \overline{\phi_{az}}}$$

Finalmente, o sinal ϕ_{em} deve ser ativo em "0" quando o dado de saída é

válido, ou seja, durante a fase ϕ_{ent} . Assim,

$$\phi_{em} = \overline{\phi_{ent}}$$

2.4.5 O MULTIPLEXADOR

Na Fig. 2.33 pode-se ver o esquema do circuito multiplexador usado neste projeto. Este circuito tem a função de colocar um conjunto de resistores "finos" em paralelo com qualquer um dos resistores "grossos".

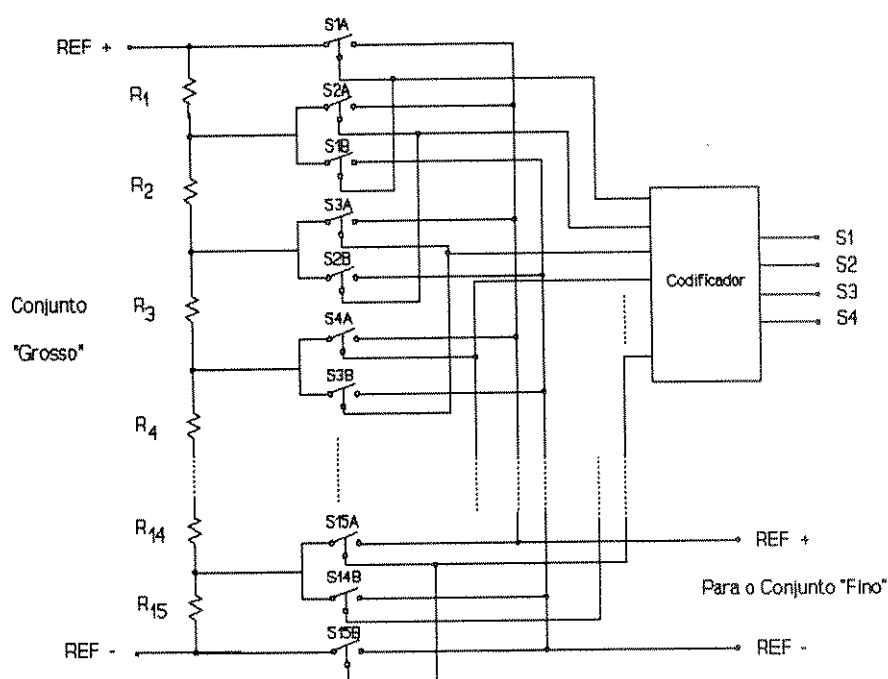


Fig. 2.33 Esquema do Multiplexador

No circuito da Fig. 2.33, se as chaves Sa1 e Sb1 estiverem ligadas, o conjunto "fino" entra em paralelo com o resistor R1 do conjunto "grosso". O multiplexador funciona de tal forma que se San e Sbn estiverem ligadas, o conjunto "fino" fica em paralelo com o resistor Rn do conjunto "grosso".

A Tab. 2.6 relaciona as saídas dos comparadores "grossos" e as chaves que devem ser ligadas. Pode-se verificar que cada combinação de entrada aciona um conjunto de duas chaves. Os conjuntos de chaves acionados são diferentes uns dos outros permitindo que uma saída acione um conjunto específico. Desta maneira, o circuito que aciona as chaves é um codificador de quatro entradas (S1 a S4) e 16 saídas (C1 a C16). As equações lógicas deste codificador, geradas a partir da Tab. 2.7, são:

$$C1 = \overline{S1} \cdot \overline{S2} \cdot \overline{S3} \cdot \overline{S4} = \overline{\overline{S1} \cdot \overline{S2} + \overline{S3} \cdot \overline{S4}} \quad (2-20)$$

$$C2 = \overline{S1} \cdot \overline{S2} \cdot \overline{S3} \cdot S4 = \overline{\overline{S1} \cdot \overline{S2} + \overline{S3} \cdot S4} \quad (2-21)$$

$$C3 = \overline{S1} \cdot \overline{S2} \cdot S3 \cdot \overline{S4} = \overline{\overline{S1} \cdot \overline{S2} + S3 \cdot \overline{S4}} \quad (2-22)$$

$$C5 = \overline{S1} \cdot S2 \cdot \overline{S3} \cdot \overline{S4} = \overline{\overline{S1} \cdot S2 + \overline{S3} \cdot \overline{S4}} \quad (2-24)$$

$$C6 = \overline{S1} \cdot S2 \cdot \overline{S3} \cdot S4 = \overline{\overline{S1} \cdot S2 + \overline{S3} \cdot S4} \quad (2-25)$$

$$C7 = \overline{S1} \cdot S2 \cdot S3 \cdot \overline{S4} = \overline{\overline{S1} \cdot S2 + S3 \cdot \overline{S4}} \quad (2-26)$$

$$C8 = \overline{S1}.S2.S3.S4 = \overline{\overline{S1}.S2 + \overline{S3}.S4} \quad (2-27)$$

$$C9 = S1.\overline{S2}.\overline{S3}.\overline{S4} = \overline{\overline{S1}.\overline{S2} + \overline{S3}.\overline{S4}} \quad (2-28)$$

$$C10 = S1.\overline{S2}.\overline{S3}.S4 = \overline{\overline{S1}.\overline{S2} + \overline{S3}.S4} \quad (2-29)$$

$$C11 = S1.\overline{S2}.S3.\overline{S4} = \overline{\overline{S1}.\overline{S2} + S3.\overline{S4}} \quad (2-30)$$

$$C12 = S1.\overline{S2}.S3.S4 = \overline{\overline{S1}.\overline{S2} + S3.S4} \quad (2-31)$$

$$C13 = S1.S2.\overline{S3}.\overline{S4} = \overline{\overline{S1}.S2 + \overline{S3}.\overline{S4}} \quad (2-32)$$

$$C14 = S1.S2.\overline{S3}.S4 = \overline{\overline{S1}.S2 + \overline{S3}.S4} \quad (2-33)$$

$$C15 = S1.S2.S3.\overline{S4} = \overline{\overline{S1}.S2 + S3.\overline{S4}} \quad (2-34)$$

$$C16 = S1.S2.S3.S4 = \overline{\overline{S1}.S2 + S3.S4} \quad (2-35)$$

As chaves do multiplexador devem apresentar baixa resistência de forma que não haja queda de tensão significativa sobre elas. Ao mesmo tempo as chaves não podem ser muito grandes para não ocuparem muita área nem carregarem demais o codificador. Com isto chegou-se a:

$$W_p = 5\mu$$

$$L_p = 2\mu$$

$$W_n = 5\mu$$

$$L_n = 5\mu$$

S1	S2	S3	S4	Chaves Ligadas
0	0	0	0	Ca1 Cb1
0	0	0	1	Ca2 Cb2
0	0	1	0	Ca3 Cb3
0	0	1	1	Ca4 Cb4
0	1	0	0	Ca5 Cb5
0	1	0	1	Ca6 Cb6
0	1	1	0	Ca7 Cb7
0	1	1	1	Ca8 Cb8
1	0	0	0	Ca9 Cb9
1	0	0	1	Ca10 Cb10
1	0	1	0	Ca11 Cb11
1	0	1	1	Ca12 Cb12
1	1	0	0	Ca13 Cb13
1	1	0	1	Ca14 Cb14
1	1	1	0	Ca15 Cb15
1	1	1	1	Ca16 Cb16

Tabela 2.7 - Lógica do Multiplexador

O esquema final do multiplexador está mostrado na Fig. 2.34. Os blocos lógicos empregados no codificador do multiplexador são os mesmos usados no codificador da saída dos comparadores.

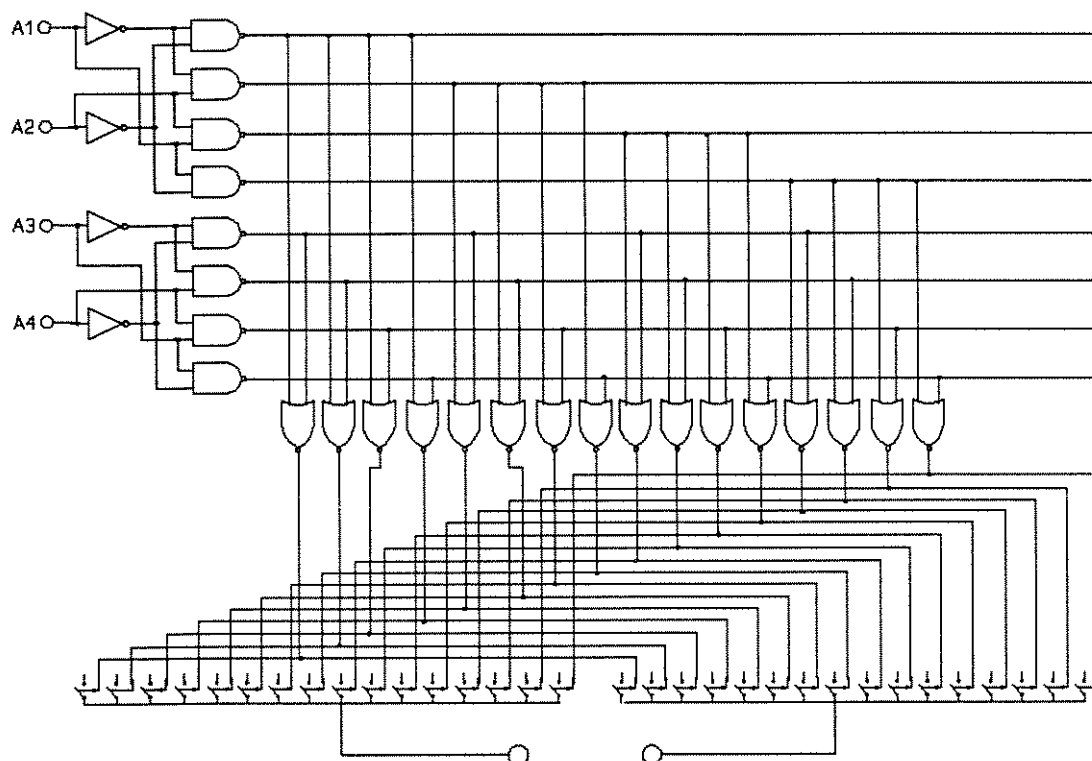


Fig. 2.34 O Diagrama do Multiplexador

2.4.6 OS REGISTRADORES DE SAÍDA

Os registradores de saída foram implementados com "latches", cujo circuito pode ser visto na Fig. 2.32. O seu funcionamento é elementar e não merece uma discussão. Este mesmo circuito integra os comparadores já analisados anteriormente.

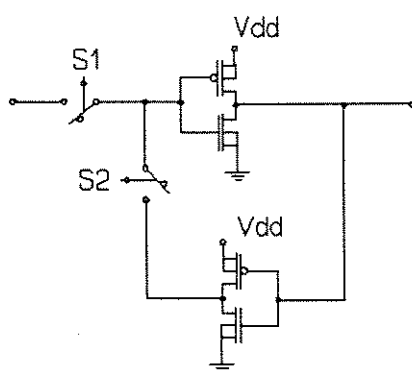


Fig. 2.35 O Circuito "Latch"

2.4.7 O "PAD" ANALÓGICO DE SAÍDA

A utilização do "pad" analógico de saída ("buffer") é necessária para que se possa exteriorizar os sinais de saída do multiplexador V_{ref+} e V_{ref-} que são analógicos, pois não se integrou os "sample-and-hold".

Lastimavelmente, o CTI não dispunha de um "pad" analógico, o que fez com que se optasse por se modificar um dos amplificadores operacionais usados para o estudo dos circuitos "sample-and-hold", ligá-lo como um "buffer" não inversor e utilizá-lo como "pad" analógico.

A Fig. 2.36 mostra o circuito do "pad" utilizado. Neste circuito, V_i é a tensão analógica de entrada, V_o é a tensão de saída do "pad" e V_r é uma tensão de 2.5V que habilita o "pad".

O circuito utilizado está longe de ser o ideal para a aplicação, pois só

opera como buffer entre 1V e 4,4V. Entretanto, servirá para se testar o conversor dentro desta faixa de tensões. O "pad" necessário para o conversor deveria ter entrada e saída "rail-to-rail", funcionando como "buffer" entre 0 e 5V.

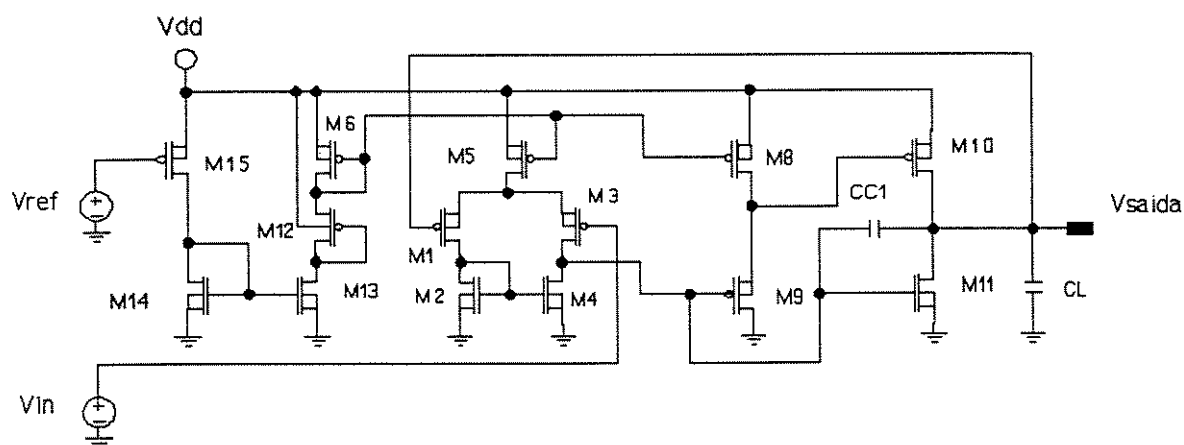


Fig. 2.36 O "Pad" Analógico de Saída

A Tab. 2.8 mostra os tempos de propagação mais críticos, obtidos na transição de baixo para alto de um sinal.

	Caso Típico	Pior Caso de Velocidade	Pior caso de Potência	Pior "0"	Pior "1"
Tempo (nS)	22.0	34.0	22.0	36.0	14.0

Tabela 2.8 Temporização do "Pad" Analógico

2.4.8 "SAMPLE-AND-HOLD" E REFERÊNCIA $V_{dd}/2$

Estes circuitos, por demandarem tempo razoável de projeto e se tratar de um "chip" teste, vão ser implementados externamente e não serão descritos neste trabalho, embora, como já visto, os sinais para ativá-los sejam exteriorizados na implementação do circuito integrado.

2.5 CONSIDERAÇÕES SOBRE O "LAYOUT" DO CONVERSOR

2.5.1 O COMPARADOR

Na execução do "layout" do comparador, alguns cuidados foram tomados com relação ao estágio de entrada:

- Os transistores desse estágio foram "casados" de forma que tudo o que um transistor "sentir" o outro irá "sentir" também.
- Foram colocados contatos de substrato para se evitar a ocorrência de "latch-up";
- O comparador foi estruturado de maneira a formar uma célula facilmente reproduzida para se encaixar uma à outra, formando um conjunto de comparadores;
- As trilhas por onde circulam os sinais de "clock" têm largura mínima para se reduzir o carregamento do gerador de fases;
- As trilhas dedicadas às referências de $V_{dd}/2$ têm o dobro da largura mínima;
- As trilhas que conduzem os sinais de entrada têm o triplo da largura mínima para se reduzir a resistência ôhmica;
- Os capacitores são de metal sobre polissilício.

A) CÁLCULO DAS DIMENSÕES DOS CAPACITORES

Concluiu-se que o capacitor de entrada poderia ser igual ao do amplificador, de valor baixo em torno de 0.2 pF. A capacitância metal-poli (típica) para o processo ES2 é 34.5 $\mu\text{F}/\text{m}$; logo, a área desses capacitores será:

$$A = \frac{0.2 \times 10^{-12} [F]}{34.5 \times 10^{-6} [F/m^2]} = 5.797,1 \mu\text{m}^2$$

Ao se conectar o capacitor ao circuito, tomou-se o cuidado de ligar o terminal da placa de metal ao gate do transistor de entrada e o terminal da placa de poli às chaves de entrada. Este cuidado deve ser tomado pois, além da capacitância poli-metal desejada, há uma capacitância poli-substrato que, se ligada ao gate do transistor de entrada, aumentaria a capacitância parasita de entrada.

O valor das capacitâncias, recalculado após a execução do "layout", foi de:

$$C_{\text{ent}} = C_{\text{amp}} = 0,2006261 \text{ pF}$$

B) CÁLCULO DOS RESISTORES DE REFERÊNCIA

Os resistores foram construídos de polissilício e alguns cuidados foram tomados:

- Tudo o que um resistor "enxergar", o mesmo deverá ocorrer com os demais;
- Não foram construídos com largura mínima para se compensar os efeitos das imprecisões de fabricação;

- Foram construídos dentro de um poço para se evitar que injetem cargas nos comparadores.

As simulações conduziram a valores de resistência em torno de 750 Ohms. A resistência típica do polissilício por quadrado é de 22 Ohms. Assim, nesta situação, o número de quadrados do resistor seria:

$$\square = \frac{750}{22} = 34,091 = \frac{\text{comprimento do resistor}}{\text{largura do resistor}}$$

Fazendo-se a largura do resistor igual a 10 μm , tem-se:

$$\text{comprimento do resistor} = 341 \mu\text{m}$$

2.5.2 OS OUTROS BLOCOS

Os demais blocos (codificador, gerador de fases e multiplexador) não necessitam de cuidados especiais e seus "layouts" foram executados regularmente.

2.5.3 AS ESTRUTURAS DE TESTE

A fim de se verificar o funcionamento e de se caracterizar os blocos mais

críticos construiu-se, separadamente, algumas estruturas de teste, ou sejam:

- Um conjunto de resistores com pontos de teste de $20 \times 20 \mu\text{m}$ em cada ponto de união de dois resistores e extremidades ligadas a "pads" externos;
- Um comparador com "pads" nas entradas e na saída;
- Um comparador com "pads" nas entradas e na saída e pontos de teste de $20 \times 20 \mu\text{m}$ em todos os pontos internos de interesse.

Estes circuitos foram feitos com alimentação, "clocks" e referências independentes do conversor e uns dos outros. O procedimento permitirá que uma estrutura possa ser testada mesmo que haja problemas com o conversor ou com outra estrutura.

2.6 RESULTADOS E CONCLUSÕES

No projeto de conversores de alta velocidade, o comparador apresenta elevada importância. Ele e os resistores da referência são os principais responsáveis pela qualidade do conversor. Por isso, receberam uma especial atenção durante todo o desenvolvimento do projeto.

Um dos maiores problemas encontrados no projeto de comparadores para conversores de precisão é a eliminação do "offset" de entrada. A técnica de auto-zero utilizada na sua eliminação é bastante razoável nos comparadores CMOS. No entanto, esta técnica não elimina por completo os efeitos do "offset" de entrada.

No projeto de comparadores CMOS "auto-zerados", o capacitor de auto-zero deve ser grande o suficiente para não perder cargas para as capacitâncias parasitas e, em contrapartida, pequeno o suficiente para não comprometer a velocidade de operação do comparador.

O estágio de entrada é o principal responsável pela precisão do comparador e seus transistores devem ser projetados de tal forma a não comprometerem o funcionamento do circuito frente às variações provocadas pela difusão.

No caso deste comparador, escolheu-se um circuito de auto-zero bastante simples, eficiente e integrável em pequena área pois a precisão do comparador exigida pelo projeto é de 1/4 do LSB (4,8 mV para sinais de entrada entre 0 e 5V) não é crítica. Entretanto, no caso de conversores com maior número de bits, deve-se estudar técnicas mais sofisticadas de auto-zero.

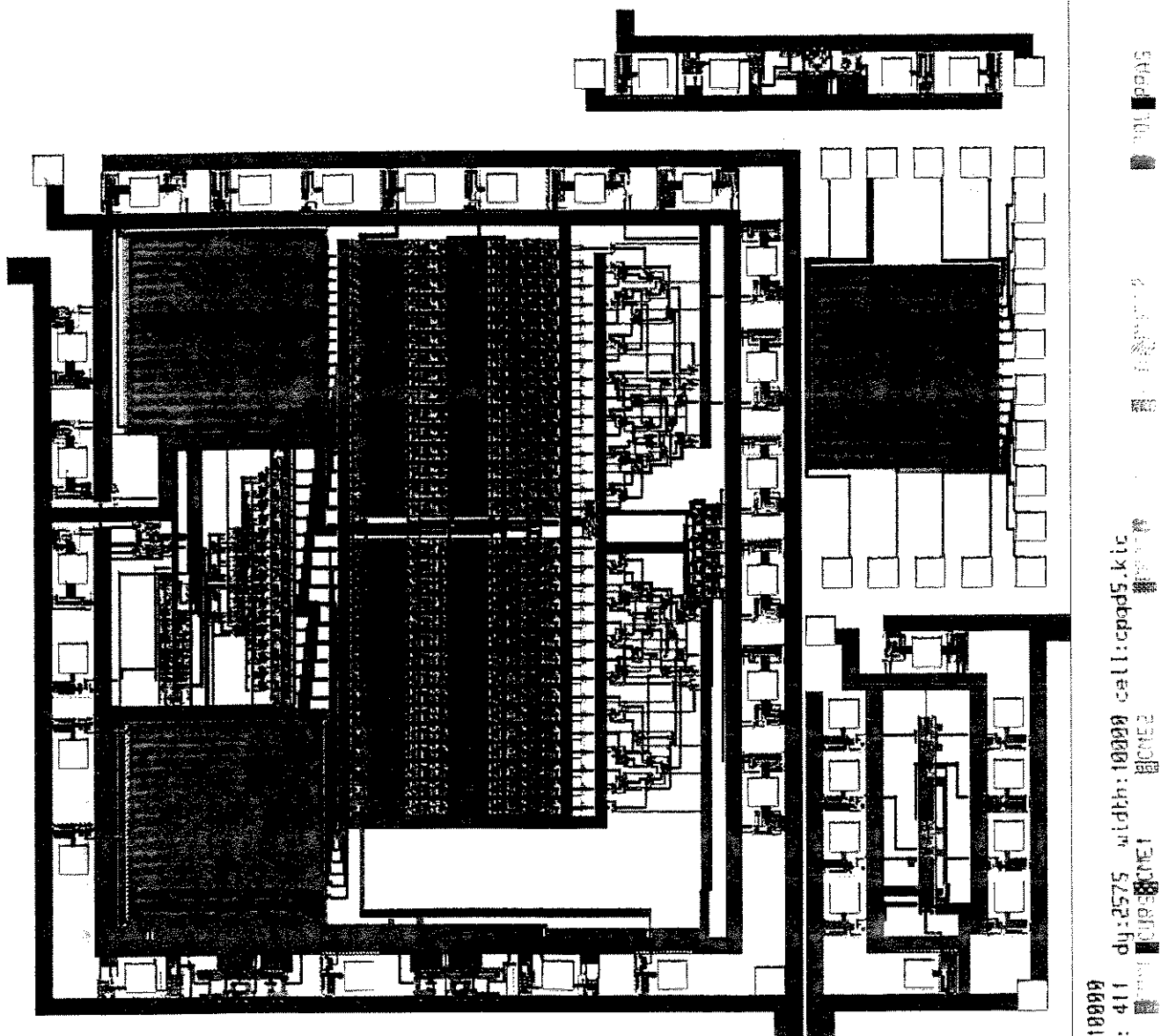
As simulações efetivadas comprovaram a alta performance do conversor e principalmente a do comparador, que atingiram com razoável folga, as especificações desejadas. Assim, fica assegurada a viabilidade da implementação do bloco com grandes chances de sucesso.

O "layout" do conversor foi elaborado visando-se minimizar a área integrada como função do compromisso de utilizá-lo como célula de circuitos VLSI pois os resultados atingidos garantem aplicações desta natureza.

A Fig. 2.37 traz uma "hardcopy" do circuito difundido, onde as estruturas discutidas no projeto podem ser identificadas com alguma facilidade. Como função de

Edit
 OR
 Save
 Write
 Attr
 Insta
 Selec
 Propy
 RDraw
 Expand
 Peak
 Pan
 Zoom
 Window
 View
 Last
 Grid
 Snap
 Boxes
 Wires
 Width
 Pullg
 DOut
 FLASH
 ARC
 Label
 Undo
 ORC
 SHERR
 Rules
 RData
 Abort
 DESug

Fig.2.37 "Hardcopy" do Conversor



uma simulação elétrica extensiva, não houve necessidade de se proceder a uma simulação lógica do conversor. Entretanto, o "layout" foi submetido a um programa DRC sem que se constatasse qualquer falha na execução.

A Fig. 2.38 é uma fotomicrografia do circuito do conversor projetado e submetido aos testes de caracterização, que serão descritos na sequência. As dimensões do conversor, incluindo as estruturas de testes, é de 2,7 mm x 2,7 mm que corresponde a uma área de 7,3 mm².

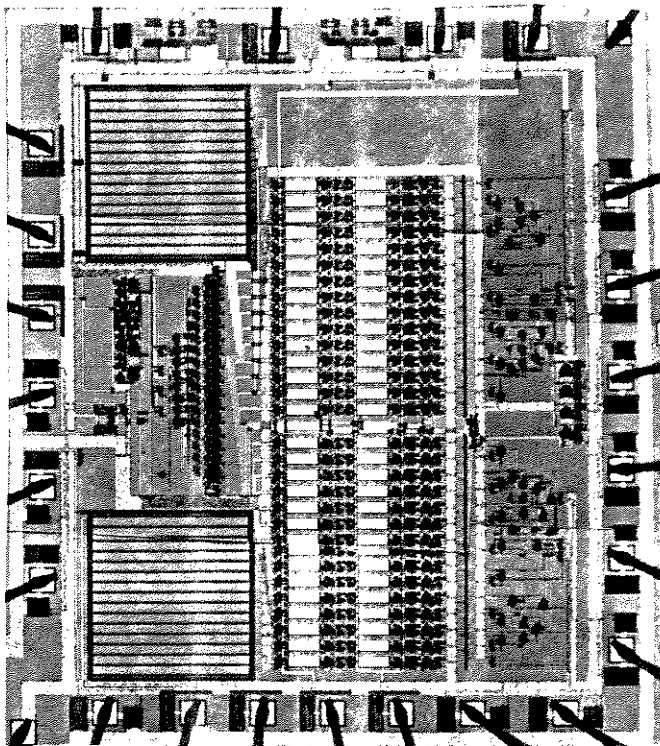


Fig. 2.38 Fotomicrografia do Conversor A/D

Os testes do Conversor A/D tiveram início com a verificação da precisão das tensões de referência obtidas através dos divisores resistivos. Um dos bancos resistivos do integrado, configurado como indicado na Fig. 2.39 e formado por resistores de 750 Ohms, foi submetido a uma tensão de 5V.

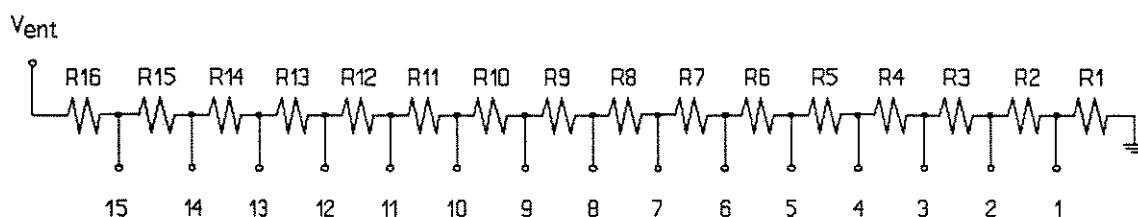


Fig. 2.39 O Divisor Resistivo

O resultado esperado era o de que cada ponto apresentasse uma variação de tensão de 312,5 mV em relação ao anterior. A Tab.2.8 ilustra os resultados obtidos em medidas realizadas em oito amostras diferentes.

Os testes do comparador foram realizados gerando-se externamente os sinais de controle já discutidos anteriormente. Foram aplicadas ainda tensões de 10 mV abaixo e acima de uma tensão "threshold" fixada em 2,5 V, que correspondia à situação de sua pior performance.

Partindo-se de uma frequência de amostragem de 500 KHz, foi-se elevando a mesma até se perceber a saturação do comparador em torno de 1,1 MHz, limitando a sua operação em um valor abaixo ao conseguido na simulação.

No teste do conversor completo, utilizou-se uma fonte de tensão de entrada para se gerar incrementos de tensão que possibilitassem a verificação do

Blocos CMOS de Alta Performance para Aplicações em VLSI

funcionamento de todos os bits.

PINO	TENSÃO EM VOLTS NAS AMOSTRAS							
	A1	A2	A3	A4	A5	A6	A7	A8
1	0,309	0,309	0,310	0,310	0,310	0,309	0,310	0,309
2	0,621	0,621	0,621	0,621	0,621	0,620	0,621	0,619
3	0,932	0,932	0,933	0,934	0,933	0,933	0,933	0,931
4	1,245	1,245	1,246	1,247	1,246	1,245	1,246	1,244
5	1,557	1,558	1,559	1,560	1,558	1,557	1,559	1,556
6	1,869	1,869	1,870	1,871	1,870	1,869	1,870	1,868
7	2,179	2,181	2,181	2,182	2,180	2,179	2,180	2,179
8	2,491	2,492	2,493	2,493	2,490	2,491	2,492	2,490
9	2,804	2,805	2,806	2,806	2,810	2,804	2,805	2,803
10	3,116	3,118	3,116	3,119	3,120	3,116	3,118	3,116
11	3,427	3,428	3,429	3,428	3,429	3,427	3,428	3,427
12	3,739	3,739	3,741	3,740	3,743	3,739	3,740	3,738
13	4,051	4,051	4,052	4,052	4,051	4,051	4,052	4,050
14	4,362	4,363	4,364	4,363	4,363	4,364	4,364	4,363
15	4,674	4,675	4,675	4,675	4,675	4,675	4,675	4,679

Tab. 2.8 Comportamento do Divisor Resistivo

Foi constatado que somente os bits menos significativos apresentavam alterações e os mais significativos permaneciam em estado alto. Um levantamento fotográfico posterior do circuito e uma análise comparativa entre o layout e o esquemático, indicaram a ausência de uma ligação de terra não realizada e a falta da conexão do sinal de auto-zero nos bits mais significativos.

Várias tentativas foram realizadas para se acessar o chip e, através de ponteiras, efetivar a conexão necessária. Entretanto, a passivação de nitreto de silício e a não existência de plasma no CTI, inviabilizaram o acesso correto.

O ataque químico que se procedeu, com certeza alterou o comportamento elétrico do circuito. Além disso, a utilização de ponteiras com ultrassom que poderiam permitir a abertura da passivação para acesso às trilhas, destruía as conexões devido as suas proximidades.

Assim, partiu-se para o teste do bloco dos bits menos significativos. As medidas foram executadas visando detetar o nível de tensão que proporcionasse a mudança de um LSB na sequência binária. Tal procedimento permitiu que se determinasse o nível real de cada transição e as inerentes variações resultantes do "layout" e da não uniformidade do processo.

As saídas dos 4 bits correspondentes foram documentadas na Tab. 2.9, que indica um funcionamento excelente do conversor pois o maior erro de não linearidade diferencial registrado foi de 0,136 LSB em ambas amostras, que fica bem aquém do valor permitido de 0,25 LSB.

Como resultado da destruição de várias amostras por ataque químico,

Blocos CMOS de Alta Performance para Aplicações em VLSI

TENSÃO IDEAL DE TRANSIÇÃO (V)	SAÍDA DIGITAL	TENSÃO REAL DA TRANSIÇÃO (V)	ERRO EM LSB
0,3125	0001	0,290	- 0,072
0,6250	0010	0,605	- 0,064
0,9375	0011	0,940	+ 0,008
1,2500	0100	1,215	- 0,112
1,5625	0101	1,580	+ 0,056
1,875	0110	1,880	+ 0,016
2,1875	0111	2,220	+ 0,104
2,500	1000	2,510	+ 0,032
2,8125	1001	2,810	- 0,008
3,125	1010	3,125	- 0,032
3,4375	1011	3,430	- 0,024
3,750	1100	3,785	+ 0,112
4,0625	1101	4,090	+ 0,088
4,375	1110	4,410	+ 0,112
4,6875	1111	4,730	+ 0,136

Tabela 2.9 Erro de Não-Linearidade Diferencial

testou-se apenas duas amostras do conversor A/D. A título de ilustração, documentou-se apenas a forma de onda de saída do bit menos significativo (Fig. 2.40). Para a sua

obtenção foi aplicado à entrada, uma tensão pulsante entre 0 e 0,4V numa frequência de 1 MHz. Pode-se perceber que a saída assume os estados alto e baixo esperados.

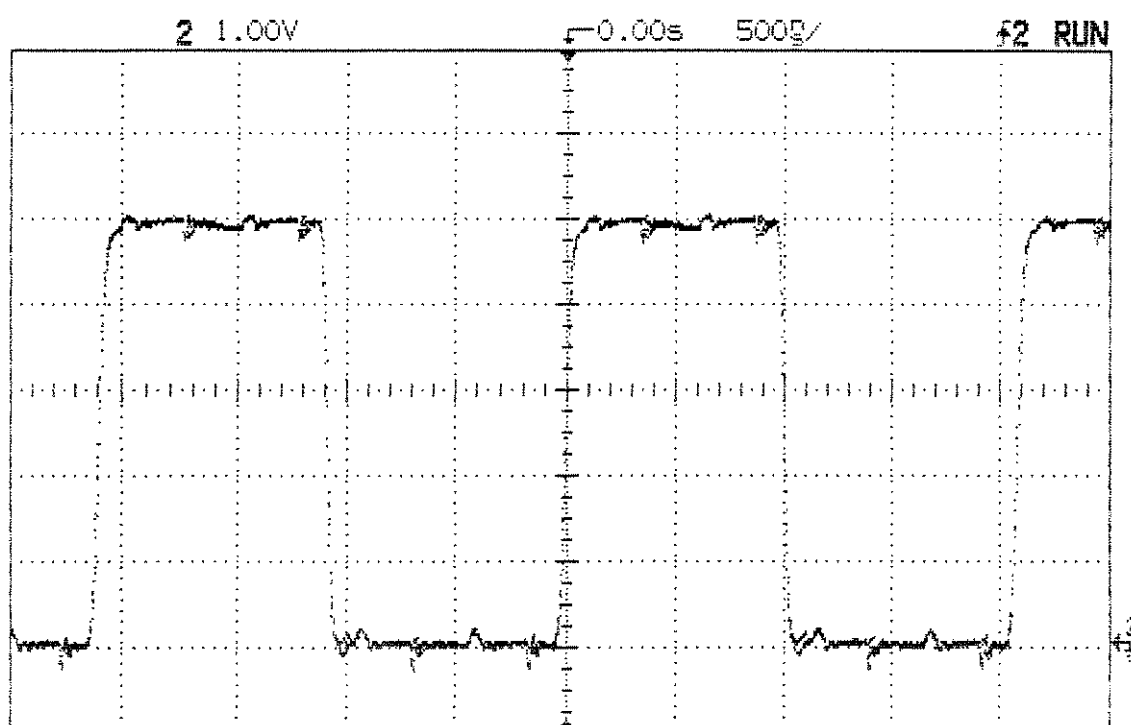


Fig. 2.40 Saída do Bit Mais Significativo

As saídas dos outros bits do conversor encontram-se disponíveis e apenas não foram adicionadas aqui pelo volume que iriam exigir.

Os resultados obtidos, embora com as limitações relacionadas anteriormente, adicionadas ao fato de não se dispor de um hardware e software compatíveis à complexidade do projeto, permitem afirmar que o conversor teve sucesso

apresentando um funcionamento excelente e dentro das especificações previstas. Melhor comprovação é a de que conseguiu-se êxito já na primeira difusão.

A impossibilidade da inclusão dos circuitos "sample-and-hold" no integrado sugere que um novo projeto deve ser estabelecido, orientado pelo autor, pois um conversor completo com as características perseguidas, simplificará de maneira significativa os projetos que o englobarem, quer em performance quanto em redução de área.

É certo que com um ferramental de projeto adequado, pode-se perseguir o projeto do subcircuito mais crítico do conversor que é o comparador. Se se conseguir minimizar o tempo de auto-zero deste componente, utilizando outras técnicas, a resposta do conversor seguramente poderá atingir frequências superiores a 20 MHz, tornando extremamente atrativa a sua implementação em qualquer sistema. Óbvio, este deverá ser o próximo passo para se conseguir conversores com maior número de bits.

Outro ponto importante de registro é que o conversor foi implementado utilizando-se um processo CMOS digital e não analógico, onde seguramente teria performance extremamente superior. Entretanto, esta postura conduziria a limitações pois o fato mais atrativo é o de não se exigir qualquer processo especial.

CAPÍTULO 3

OS MULTIPLICADORES DIGITAIS

3.1 CONSIDERAÇÕES SOBRE MULTIPLICADORES

O multiplicador é uma parte essencial de qualquer circuito de processamento de sinal digital. Na grande maioria das operações de processamento, como no caso de correlação, convolução, filtragem e análise de frequência, há o envolvimento com alguma multiplicação. Principalmente por este motivo, o multiplicador digital foi um dos blocos escolhidos para integrar este trabalho.

De uma forma resumida, pretende-se introduzir algumas considerações sobre os multiplicadores série e paralelo, na sua forma mais simples sendo que os algoritmos apresentados servirão apenas para ilustrar métodos de projeto de diferentes

células, de forma a se encaixarem dentro de uma estrutura maior de multiplicação.

A forma mais elementar de multiplicação consiste na formação do produto de dois números binários positivos. A computação poderia ser efetivada através da tradicional técnica de adições e deslocamentos sucessivos, exaustivamente ilustrados na literatura, em que cada adição é condicionada a um dos bits do multiplicador.

Exemplificando, a multiplicação de dois números binários positivos como 1101 (equivalente ao decimal 13) e 0101 (equivalente ao decimal 5), poderia ser realizada como:

$$\begin{array}{rcccccc} & & & & 1 & 1 & 0 & 1 \\ & & & & 0 & 0 & 0 & 0 \\ & & 1 & 1 & 0 & 1 & & \\ 0 & 0 & 0 & 0 & & & & \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & 1 & = 65 \text{ (decimal)} \end{array}$$

Pode-se perceber, no exemplo acima, que os processos de multiplicação consistem basicamente de duas etapas:

- a) cálculo do produto parcial;
- b) acumulação do produto parcial deslocado.

A multiplicação binária é equivalente à operação lógica "E". Assim, o cálculo dos produtos parciais consiste em se efetuar a lógica "E" entre o multiplicando e o equivalente bit do multiplicador. Na sequência, procede-se à acumulação dos

produtos parciais devidamente deslocados que, posteriormente, serão somados para se obter o resultado final da multiplicação.

Há várias técnicas possíveis de serem aplicadas para se realizar a multiplicação indicada acima. Geralmente, a opção por uma técnica específica prende-se a fatores como velocidade, precisão numérica, área para integração e outros.

Como regra básica, os multiplicadores podem ser classificados, pela forma com que os dados são acessados, em:

- Multiplicadores Série;
- Multiplicadores Série/Paralelo;
- Multiplicadores Paralelo.

Na sequência, apresentar-se-á resumidamente uma estrutura básica de cada tipo de multiplicador.

3.2 O MULTIPLICADOR SÉRIE

A estrutura mais elementar de um multiplicador está indicada na Fig. 3.1 [6]. Ela utiliza o algoritmo das adições sucessivas e emprega, para tanto, apenas um somador completo, um circuito lógico "E", um elemento de atraso (tipo flip-flop) e um registrador série-paralelo.

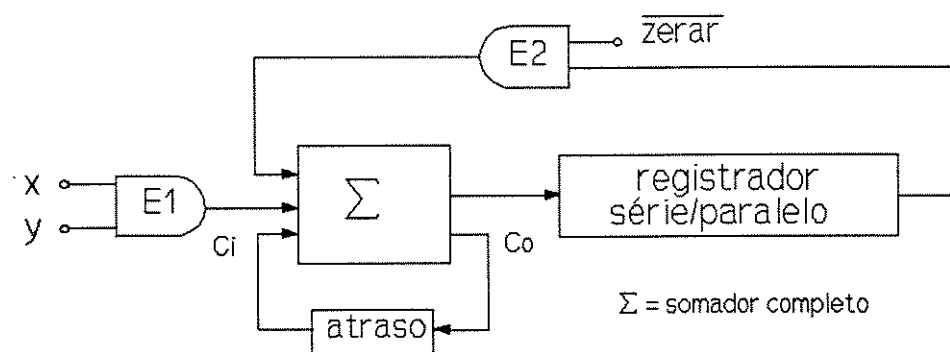


Fig. 3.1 Estrutura Básica de um Multiplicador Série

Os números X e Y são apresentados serialmente ao circuito em diferentes velocidades, de forma a compatibilizar a operação "E" entre a palavra do multiplicando e cada bit específico da palavra do multiplicador. Em outras palavras, a palavra do multiplicando passa por uma operação "E" serial com cada bit particular do multiplicador, gerando um produto parcial.

Cada produto parcial, calculado da maneira descrita acima e deslocado de um bit para a esquerda, sofre uma operação de adição série com as somas parciais anteriormente obtidas e armazenadas no registrador série-paralelo.

O circuito E2, indicado na Fig. 3.1, é usado para permitir que o registrador possa ser zerado no início de cada ciclo de multiplicação.

Em termos de velocidade de operação, o produto final pode ser obtido com $M \times N$ intervalos de "clock" do multiplicando, onde M é o número de bits do multiplicador e N o do multiplicando.

3.3 O MULTIPLICADOR SÉRIE-PARALELO

O multiplicador série-paralelo pode facilmente ser implementado com base nas considerações apresentadas para o multiplicador série. A Fig. 3.2 ilustra a implementação de uma estrutura modular bastante simples, que pode ser facilmente modificada para a de um sistema tipo "pipeline".

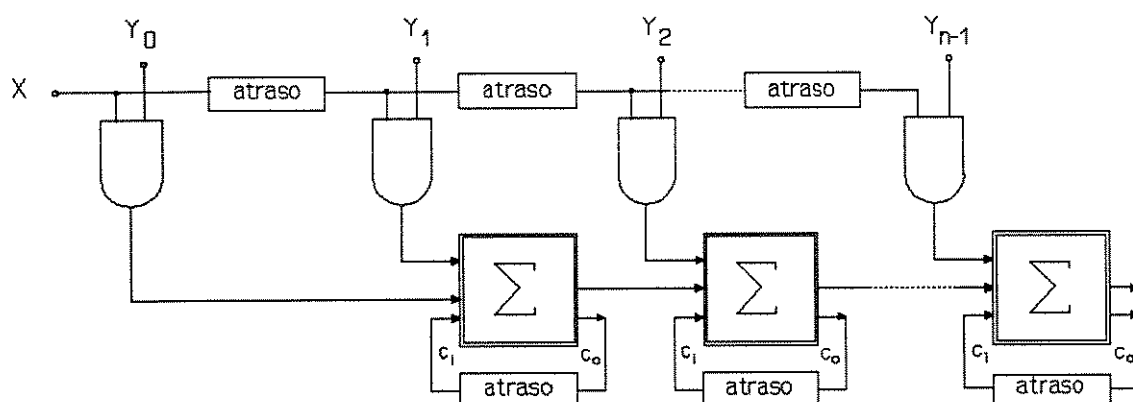


Fig. 3.2 Estrutura de um Multiplicador Série-Paralelo

Na estrutura mostrada, a multiplicação é realizada através de adições sucessivas das colunas da matriz de produtos parciais deslocados. Como exemplo desta matriz, no caso da multiplicação entre os números binários $Y = Y_3Y_2Y_1Y_0$ e $X = X_2X_1X_0$, ter-se-ia o resultado indicado na Tab. 3.1.

Assim como nos sistemas seriais em que o deslocamento dos números é conseguido com um elemento de atraso (equivalente ao tempo de um bit), o multiplicador é sucessivamente deslocado à direita e gatilha o bit apropriado do multiplicando.

		X_0Y_3	X_0Y_2	X_0Y_1	X_0Y_0
	X_1Y_3	X_1Y_2	X_1Y_1	X_1Y_0	
X_2Y_3	X_2Y_2	X_2Y_1	X_2Y_0		

Tabela 3.1 - Matriz de Produtos Parciais Deslocados

Os estágios do multiplicando precisam, após o atraso, ficar alinhados na mesma coluna da matriz de produtos parciais deslocados, quando então são somados para formarem o bit do produto requerido da coluna em questão.

A estrutura apresentada requer $M+N$ ciclos de "clock" para produzir um produto. A principal desvantagem é que a frequência máxima é limitada pela propagação através do arranjo de somadores.

3.4 O MULTIPLICADOR PARALELO

O multiplicador paralelo é baseado no fato de que os produtos parciais, característicos da multiplicação, podem ser computados em paralelo de uma forma independente. Como exemplo, seja os inteiros binários X e Y sem sinal:

$$X = \sum_{i=0}^{M-1} X_i 2^i \quad Y = \sum_{j=0}^{N-1} Y_j 2^j$$

O produto pode ser encontrado como:

$$PRODUTO\ X.Y = \sum_{i=0}^{M-1} X_i 2^i \sum_{j=0}^{N-1} Y_j 2^j = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (X_i Y_j) 2^{i+j} \quad (3-1)$$

ou,

$$P_k = \sum_{k=0}^{M+N-1} P_k 2^k$$

P_k são os termos do produto parcial denominados somas parciais. Pode-se perceber pela Eq.(3-1) que há MN somandos parciais, que são produzidos em paralelo por um conjunto de MN portas "E". Para números de quatro bits, como exemplo, a expressão mostrada acima pode ser expandida como:

				X ₃	X ₂	X ₁	X ₀	Multiplicando
				Y ₃	Y ₂	Y ₁	Y ₀	Multiplicador
				X ₃ Y ₀	X ₂ Y ₀	X ₁ Y ₀	X ₀ Y ₀	
				X ₃ Y ₁	X ₂ Y ₁	X ₁ Y ₁	X ₀ Y ₁	
				X ₃ Y ₂	X ₂ Y ₂	X ₁ Y ₂	X ₀ Y ₂	
				X ₃ Y ₃	X ₂ Y ₃	X ₁ Y ₃	X ₀ Y ₃	
P ₇	P ₆	P ₅	P ₄	P ₃	P ₂	P ₁	P ₀	Produto Final

Um multiplicador $N \times N$ requer $N(N-1)$ somadores completos, N meio

somadores e N^2 portas "E". O atraso em pior caso, associado ao multiplicador, é igual a $(2N+1)\Phi_s$ onde Φ_s é o atraso do somador em pior caso.

A Fig. 3.3 mostra uma célula que pode ser usada para se construir um multiplicador paralelo. O termo X_i propaga-se verticalmente enquanto que o termo Y_i o faz horizontalmente. Os produtos parciais que chegam à célula entram pelo centro de seu topo e o "carry" de entrada, pela direita do topo.

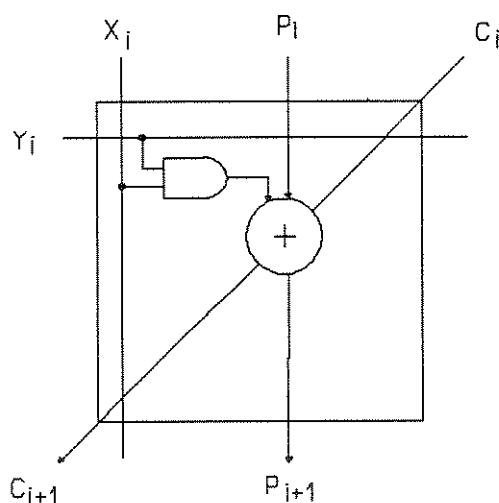


Fig. 3.3 Célula de um Multiplicador Paralelo

Prosseguindo, a lógica "E" é realizada dentro da célula enquanto que a SOMA é levada à próxima célula pelo centro da parte inferior. Finalmente, o "carry" de saída é enviado à próxima célula pela extremidade inferior esquerda.

A Fig. 3.4 mostra o arranjo de um multiplicador com os produtos parciais numerados. O arranjo pode ser desenhado em forma quadrada como na Fig. 3.5, que é a forma mais conveniente para implementação.

O projeto da célula deste multiplicador é relativamente direto, com a principal atenção voltada ao somador. Um somador com iguais tempos de propagação de soma e de "carry" é vantajoso, pois o tempo da multiplicação, em pior caso, depende de ambas as trajetórias.

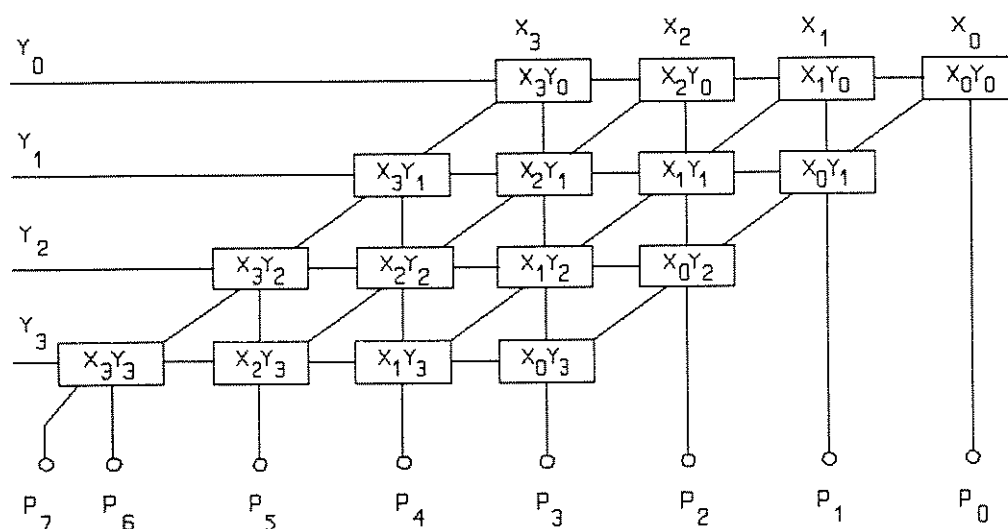


Fig. 3.4 Arranjo de um Multiplicador Paralelo

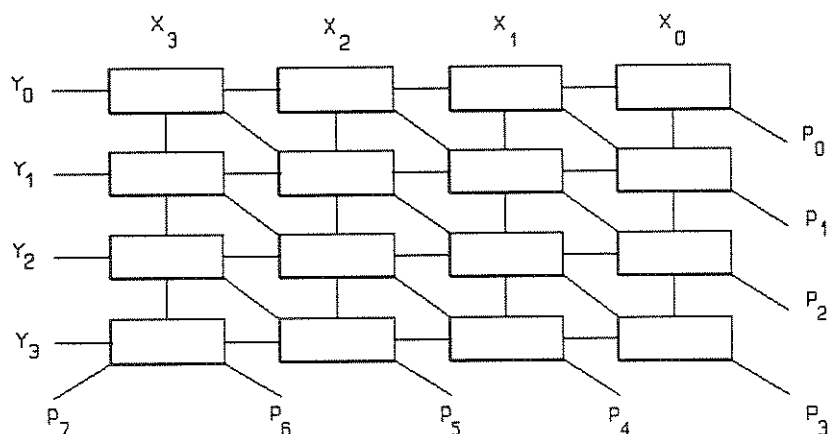


Fig. 3.5 Arranjo Mais Conveniente de um Multiplicador Paralelo

O tempo necessário ao cálculo do produto é determinado primariamente pelo atraso de propagação dos sinais dentro do arranjo. Assim, o produto é gerado muito mais rapidamente nesta estrutura quando comparado às estruturas dos multiplicadores série e série/paralelo. Entretanto, o preço pago pelo aumento da velocidade é que a área requerida no integrado é proporcional a N^2 , onde N é o comprimento da palavra do multiplicando e do multiplicador.

As aproximações simples apresentadas neste texto para a multiplicação objetivaram ilustrar a forma com que sistemas maiores podem ser construídos a partir de componentes menores. As arquiteturas neste nível são diversas e podem ser encontradas facilmente na literatura.

O projeto de um multiplicador por hardware depende de restrições que devem ser satisfeitas ao nível arquitetural, isto é, se se deseja uma capacidade de se manipular um grande número de bits em lugar de um pequeno atraso e assim por diante.

Neste trabalho, optou-se pelo projeto de um multiplicador em que a densidade de circuitos e a "latência" foram considerados os pontos mais importantes de mérito a serem minimizados. Com estas considerações, os multiplicadores série/paralelo e série não poderiam ser considerados devido a sua alta latência. Entretanto, decidiu-se que seria projetado um multiplicador série com base no mesmo algoritmo escolhido para a implementação do paralelo.

Dentre todos os esquemas paralelos, o multiplicador matricial é preferido pela sua regularidade extremamente alta. Por outro lado, um multiplicador matricial básico, como o de Baugh e Wooley [3], que trabalha por complemento de

dois, exige muita área quando se precisa processar operandos com mais de 16 bits. Um multiplicador de 32 bits, por exemplo, ocupa em torno de 5 a 6 mm² (tecnologia 2 μ m), dependendo do estilo de projeto lógico (dinâmico ou estático).

A proposta do projeto exigia um multiplicador que não fosse apenas pequeno, mas que também fosse rápido. Se se considerasse a velocidade como único parâmetro, várias estruturas poderiam ser usadas. As árvores de WALLACE e os contadores paralelos de DADDA [4] estão entre os mais conhecidos. Por outro lado, sabe-se também que, devido a sua estrutura em árvore, não são propícios para uma implementação regular.

O multiplicador em que se está interessado para atender as necessidades é um que possa processar eficientemente operando com até 64 bits. Assim, não serão necessárias aproximações muito exóticas como as apresentadas em [5] e [7], que usualmente conduzem a estruturas mais comuns quando são considerados operandos com mais de 64 ou 128 bits.

Uma outra importante consideração é que o sistema numérico deve entrar como um dado na escolha. O sistema numérico posicional com raiz $r=2$, que corresponde ao sistema binário normal, é assumido. Mais precisamente, o complemento de dois foi adotado.

Interessantes resultados poderiam ser conseguidos usando-se diferentes sistemas numéricos tal como o sistema numérico residual [8], por exemplo. Além disso, um multiplicador apresentando regularidade, tempo reduzido de execução e pequena área, seria interessante também quando sistemas numéricos não-posicionais são considerados.

Com as considerações acima, para o projeto do multiplicador paralelo deste trabalho, optou-se pelo algoritmo modificado proposto por Booth, que conduz a uma arquitetura avançada. O mesmo algoritmo também foi adotado para a implementação do multiplicador série. Assim, far-se-á na Seção seguinte, uma breve revisão do algoritmo original e do modificado.

3.5 O ALGORITMO DE BOOTH

O Algoritmo de Booth original [1] pertence à classe de "algoritmos de recodificação", ou seja, algoritmos que recodificam um dos dois operandos de forma a reduzir o número de produtos parciais a serem somados juntos.

Como ilustração, um algoritmo simples de recodificação poderia consistir na omissão de todos os zeros de um dos operando com base no fato de que não contribuem para o resultado final. Entretanto, este algoritmo conduziria a um tempo de execução variável da multiplicação.

A situação descrita acima pode ser interessante em sistemas auto-temporizados, onde um sinal de "pronto" pode anunciar o fim da operação; entretanto, é inútil nos casos em que o usuário tem que disparar o sistema no pior caso, isto é, quando o operando só tem 1. Realmente, o próprio algoritmo de Booth não é de tempo de execução constante. Entretanto, é bem mais efetivo que o algoritmo que simplesmente salta zeros, uma vez que lida com sequências de 0's e de 1's devidamente recodificados.

O algoritmo de Booth pode ser usado para se construir um multiplicador por complemento de dois em que todos os bits recebem tratamento igual. O método é equivalente a se recodificar o coeficiente com uma sequência de dígitos ternários (-1, 0, +1) com pesos binários.

A multiplicação resulta da adição de produtos parciais negativos, nulos e positivos, formados pela multiplicação de cada Y pela palavra de dado. A Tab. 3.2 lista as possibilidades que podem ocorrer. As adições devem ser feitas em aritmética por complemento de dois, usando-se extensões de sinal quando necessário. A tabela sugere ainda o uso de Y_i para se determinar uma adição ou subtração, enquanto que $Y_i \oplus Y_{i-1}$ comuta as palavras de dado.

Multiplicador Original		Multiplicador Recodificado	Conexão	Ação
Y_i	Y_{i-1}	Y_i	$Y_i \oplus Y_{i-1}$	
0	0	0	0	Adicionar zero
0	1	+1	1	Adicionar palavra de dado
1	0	-1	1	Subtrair palavra de dado
1	1	0	0	Subtrair zero

Tabela 3.2 - O Algoritmo de Booth Original

3.6 O ALGORITMO DE BOOTH MODIFICADO

Um algoritmo levemente diferente, denominado Algoritmo de Booth Modificado [10], considera estritamente grupos de bits em um dos operandos. O procedimento conduz a um maior tempo de execução, porém constante.

Uma breve análise do algoritmo será apresentada. Como ilustração do método, será considerado um multiplicador de 8 bits operando sobre os números $X = X_7, \dots, X_0$ e $Y = Y_7, \dots, Y_0$.

O processo de se multiplicar dois números binários de n bits cada é equivalente à realização manual em que se adiciona números de n bits, adequadamente deslocados, n vezes. Este método é usado nos multiplicadores matriciais.

O esquema de recodificação por par de bits do algoritmo de Booth pode ser encontrado em [10] da forma reportada na Tab. 3.3. O algoritmo diminui o número de produtos parciais a serem somados juntos, acelerando o processamento.

O processo de aceleração depende do número de bits que o algoritmo considera em cada passo. Se for considerado apenas um par, o que ocorrerá neste trabalho, o arranjo é denominado "recodificação por par de bits".

Se se utilizasse um esquema de recodificação de três bits ou mais, um significativo aumento de velocidade poderia ser conseguido com uma razoável redução de área. Entretanto, há boas razões para não se implementá-los.

No caso do esquema de recodificação por três bits, por exemplo, também se necessitaria dos termos $3 \cdot X$ e $4 \cdot X$. O tempo necessário para computá-los, especialmente quando se trata de grandes operandos, definitivamente desencoraja o

uso de tais arranjos, exceto no tocante a grandes computadores, o que não é o caso deste trabalho.

Bit i+1	Bit i	Bit i-1	Dígito Recodificado
0	0	0	0
0	0	1	+1
0	1	0	+1
0	1	1	+2
1	0	0	-2
1	0	1	-1
1	1	0	-1
1	1	1	0

Tabela 3.3 - Algoritmo de Booth Modificado

Mesmo com referência a estes computadores, para não se perder velocidade, há a necessidade de se implementar somadores extremamente rápidos, que usualmente não têm disposição regular e ocupam uma área grande, não solucionando o problema.

A aceleração, se comparada à implementação direta via multiplicador matricial, também depende de como os dois multiplicadores são implementados. De acordo com ANNARATONE e SHEN [11], se ambos usarem um somador do tipo "carry look-ahead", o multiplicador de Booth será mais rápido em torno de quarenta

por cento. Além disso, a área de integração decrescerá significativamente (em torno de quarenta por cento).

Seja agora um esquema de recodificação por par de bits. O algoritmo opera sobre um dos dois operandos, analisa pares de bits e os converte em um dígito do conjunto formado pelos números 0, +1, +2, -1 e -2. Como exemplo, se o operando Y a ser recodificado for:

0 1 1 0 1 1 1 0

o algoritmo terá, de acordo com a Tab. 3.3, a seguinte sequência recodificada:

+2 -1 0 -2

Cada dígito recodificado realiza algum processamento no outro operando X, de acordo com a Tab. 3.4.

Dígito Recodificado	Operação em X
0	Some 0 ao produto parcial
+1	Some X ao produto parcial
+2	Some 2*X ao produto parcial
-1	Subtraia X do produto parcial
-2	Subtraia 2*X do produto parcial

Tabela 3.4 - Influência do Dígito Recodificado em X

O exemplo que se segue mostrará como o algoritmo trabalha. Sejam os números binários com sinal $X = 1\ 0\ 1\ 1\ 0\ 1\ 0\ 1$ e $Y = 0\ 1\ 1\ 1\ 0\ 0\ 1\ 0$. Recodificando-se o número Y, com base na Tab. 3.3, ter-se-á :

$$0\ 1\ 1\ 1\ 0\ 0\ 1\ 0 \Rightarrow +2\ -1\ +1\ -2$$

A multiplicação completa está ilustrada na Tab. 3.5.

X = 1 0 1 1 0 1 0 1 Y = 0 1 1 1 0 0 1 0	Ação
0 0 0 0 0 0 0 0 1 0 0 1 0 1 1 0 1 1 1 1 1 1 1 0 1 1 0 1 0 1 0 0 0 0 0 1 0 0 1 0 1 1 1 1 0 1 1 0 1 0 1 0	(-2):Fazer o complemento de 2 de X e deslocar uma posição a esquerda (+1):Adicionar X deslocado duas posições a esquerda(note a extensão do sinal) (-1):Tomar o complemento de 2 de X (+2):Deslocar X uma posição a esquerda
1 1 0 1 1 1 1 0 1 0 0 1 1 0 1 0	Resultado Final de 16 Bits

Tabela 3.5 O Processo de Multiplicação Completo

Para se implementar o multiplicador de Booth, há ainda a necessidade de se resolverem dois problemas fundamentais:

- 1) Identificar as funcionalidades necessárias dentro do arranjo, de forma a se

implementar o algoritmo.

2) Devido ao problema da "extensão do sinal", o formato do multiplicador passa a ser trapezoidal em lugar de retangular. Logo, é necessário se reconstruir a forma do arranjo para um retângulo, através da análise desse problema.

A próxima Seção discutirá as funcionalidades necessárias dentro do arranjo, para se conseguir implementar o algoritmo.

3.6.1 AS FUNCIONALIDADES NECESSÁRIAS NO BLOCO

Seja $X = X_{n-1}X_{n-2}...X_1X_0$ um número de n bits. O seu complemento de dois será $X_{cd} = \bar{X} + 1$. Um número de n bits em complemento de dois pode representar valores na faixa de -2^{n-1} (1000...000) a $+2^{n-1}$ (0111...111).

Durante o processo de multiplicação, os produtos parciais são computados multiplicando-se o número binário X pelo dígito adequadamente recodificado, que pode ser 0, +1, -1 ou -2. Neste caso, o produto parcial pode estar na faixa de -2^n ($2*2^{n-1}$) a 2^n ($2*2^{n-1}$).

Do exposto acima, seriam necessários, no mínimo, $n+2$ bits para se representar o produto parcial. Entretanto, quando o complemento de dois de um número é tomado, há também uma operação de "soma 1" a ser realizada. Logo, o número máximo de bits que são realmente necessários para se representar qualquer produto parcial é $n+1$.

De acordo com a discussão anterior, todos os produtos parciais podem ser gerados por uma estrutura que requer somente elementos de multiplexagem e uma operação de soma no bit menos significativo.

A Tab. 3.6 mostra a relação entre o multiplicando $X = X_7X_6...X_0$ e o produto parcial $PP = PP_8PP_7...PP_0$. Aparentemente, precisa-se realizar uma adição extra para se levar em conta a operação de soma no LSB. Entretanto, é possível "atrasar" esta operação simplesmente usando-se uma técnica de salvar "carry" comum e um somador "carry look-ahead" na extremidade inferior do arranjo.

Dígito Recodificado	Produtos Parciais									Soma	Operação
	PP ₈	PP ₇	PP ₆	PP ₅	PP ₄	PP ₃	PP ₂	PP ₁	PP ₀		
0	0	0	0	0	0	0	0	0	0	0	
+1	X ₇	X ₇	X ₆	X ₅	X ₄	X ₃	X ₂	X ₁	X ₀	0	
-1	$\bar{X}_7$	$\bar{X}_7$	$\bar{X}_6$	$\bar{X}_5$	$\bar{X}_4$	$\bar{X}_3$	$\bar{X}_2$	$\bar{X}_1$	$\bar{X}_0$	1	Inverta X e some 1 ao LSB
+2	X ₇	X ₆	X ₅	X ₄	X ₃	X ₂	X ₁	X ₀	X ₋₁	0	Desloque X uma posição à esquerda
-2	$\bar{X}_7$	$\bar{X}_6$	$\bar{X}_5$	$\bar{X}_4$	$\bar{X}_3$	$\bar{X}_2$	$\bar{X}_1$	$\bar{X}_0$	$\bar{X}_{-1}$	1	Desloque X uma posição à esq.e some 1 ao LSB.

Tabela 3.6 Relação entre Produto Parcial e Dígitos Recodificados

Como é aparente na tabela, qualquer produto parcial pode ser

produzido por um circuito multiplexador e um somador. Logo, pode-se obter uma tabela da verdade do tipo mostrado na Tab. 3.7. A equação booleana que descreve o multiplexador, com base nesta tabela, é:

$$PP(j) = P1 \wedge X(i) \vee P2 \wedge X(i-1) \vee M1 \wedge \bar{X}(i) \vee M2 \wedge \bar{X}(i-1) \quad (3-2)$$

enquanto que a lógica do bloco de soma 1 é simplesmente,

$$Soma = M1 \vee M2 \quad (3-3)$$

Dígito Recodificado	Saída		Operação de Soma
0 (0)	PP(i)=0	para i=1,...,8	0
1 (P1)	PP(i)=X(i)	para i=1,...,8	0
2 (P2)	PP(i)=X(i-1)	para i=1,...,8	0
-1 (M1)	PP(i)= $\bar{X}(i)$	para i=1,...,8	1
-2 (M2)	PP(i)= $\bar{X}(i-1)$	para i=1,...,8	1

Notas:a)Note que X(8)=X(7)=bit de sinal do multiplicando;

b)PP(8)=bit de sinal do produto parcial

Tabela 3.7 - Tabela da Verdade para Produtos Parciais

As três células necessárias dentro do arranjo para se implementar o multiplicador de Booth estão mostradas na Fig. 3.6. Um somador completo e dois diferentes circuitos combinacionais C1 e C2 foram empregados. O circuito C1 é um

multiplexador e C2 é um gerador de soma 1.

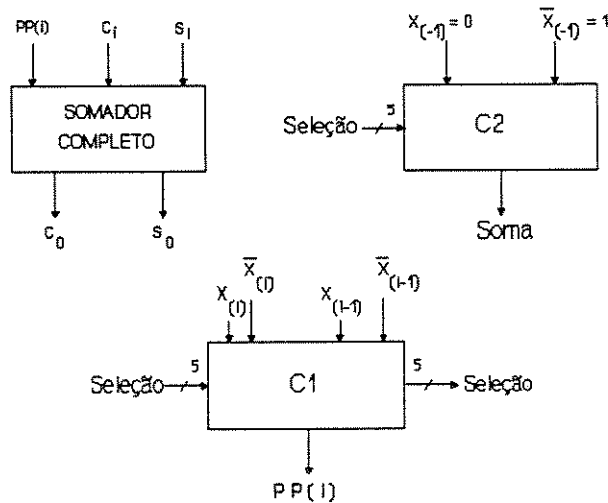
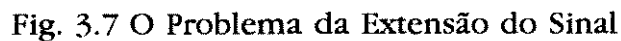


Fig. 3.6 As Três Células Básicas

3.6.2 O PROBLEMA DA EXTENSÃO DO SINAL

Das discussões anteriores, sabe-se que somente nove bits e uma operação de soma são necessários para se representar qualquer produto parcial, com o nono bit sendo usado simplesmente para se representar o sinal. Isto conduz obviamente a um multiplicador que não tem forma retangular, como mostrado na Fig. 3.7. Assim, de forma a fazer com que o arranjo volte a ter forma retangular, a extensão do sinal precisa ser agora considerada.

Há várias aproximações utilizadas na solução do problema da extensão do sinal. Particularmente interessante é a conhecida por método do sinal propagado [12



Na sequência, apresentar-se-á a aproximação utilizada no projeto do multiplicador para se resolver o problema da extensão do sinal, conhecida como "método do sinal gerado" [11].

3.6.3 MÉTODO DO SINAL GERADO

Um método bastante interessante e simples para se tratar o problema da extensão do sinal será agora apresentado. O algoritmo é discutido para um multiplicador de 8 bits, mas pode ser facilmente estendido a operandos com qualquer comprimento de palavra.

Pode-se escrever o bit do sinal do resultado de uma multiplicação [11]

como:

$$S = S_0 \sum_{i=8}^{15} 2^i + (S_1 \sum_{i=8}^{13} 2^i) * 2^2 + (S_2 \sum_{i=8}^{11} 2^i) * 2^4 + (S_3 \sum_{i=8}^9 2^i) * 2^6 \quad (3-4)$$

onde S_0 , S_1 , S_2 e S_3 são os bits de sinais dos quatro produtos parciais inerentes ao multiplicador de oito bits.

Usando-se as duas equivalências,

$$\sum_{i=j}^k = 2^j (2^{k+1-j} - 1) = 2^{k+1} - 2^j \quad (3-5)$$

e

$$S_j = 1 - \overline{S_j} \quad (3-6)$$

S fica:

$$S = (1 - \overline{S_0}) (2^{16} - 2^8) + (1 - \overline{S_1}) (2^{16} - 2^{10}) + (1 - \overline{S_2}) (2^{16} - 2^{12}) + (1 - \overline{S_3}) (2^{16} - 2^{14})$$

ou,

$$S = [4 (S_0 + S_1 + S_2 + S_3)] * 2^{16} + S_0 2^8 + S_1 2^{10} + S_2 2^{12} + S_3 2^{14} - 2^8 - 2^{10} - 2^{12} - 2^{14}$$

Finalmente,

$$S = [3 (S_0 + S_1 + S_2 + S_3)] * 2^{16} + S_0 2^8 + S_1 2^{10} + S_2 2^{12} + S_3 2^{14} + 2^{16} - (2^8 + 2^{10} + 2^{12} + 2^{14})$$

Como se tem,

$$2^{16} = \sum_{i=0}^{15} 2^i = \sum_{i=0}^7 2^i + 2^8 + 2^9 + 2^{10} + 2^{11} + 2^{12} + 2^{13} + 2^{14} + 2^{15}$$

que é igual a

$$S = 2^8 + 2^8 + 2^9 + 2^{10} + 2^{11} + 2^{12} + 2^{13} + 2^{14} + 2^{15}$$

o sinal do resultado será:

$$S = [3 (\overline{S_0} + \overline{S_1} + \overline{S_2} + \overline{S_3})] * 2^{16} + \overline{S_0} 2^8 + \overline{S_1} 2^{10} + \overline{S_2} 2^{12} + \overline{S_3} 2^{14} + 2^8 + 2^9 + 2^{11} + 2^{13} + 2^{15} \quad (3-7)$$

O primeiro termo de S é o décimo sétimo bit e pode ser ignorado. Assim, S pode ser reescrito como:

$$S = \overline{S_0} 2^8 + \overline{S_1} 2^{10} + \overline{S_2} 2^{12} + \overline{S_3} 2^{14} + 2^9 + 2^{11} + 2^{13} + 2^{15} + 2^8 \quad (3-8)$$

A Eq.(3-8) pode ser interpretada como:

1. Complemente o bit de sinal de cada produto parcial.
2. Some 1 à esquerda do bit de sinal de cada produto parcial.
3. Some 1 ao nono (N+1) bit do primeiro produto parcial.

Um exemplo deste método está ilustrado na Fig. 3.8, para uma multiplicação de dois números binários de oito bits.

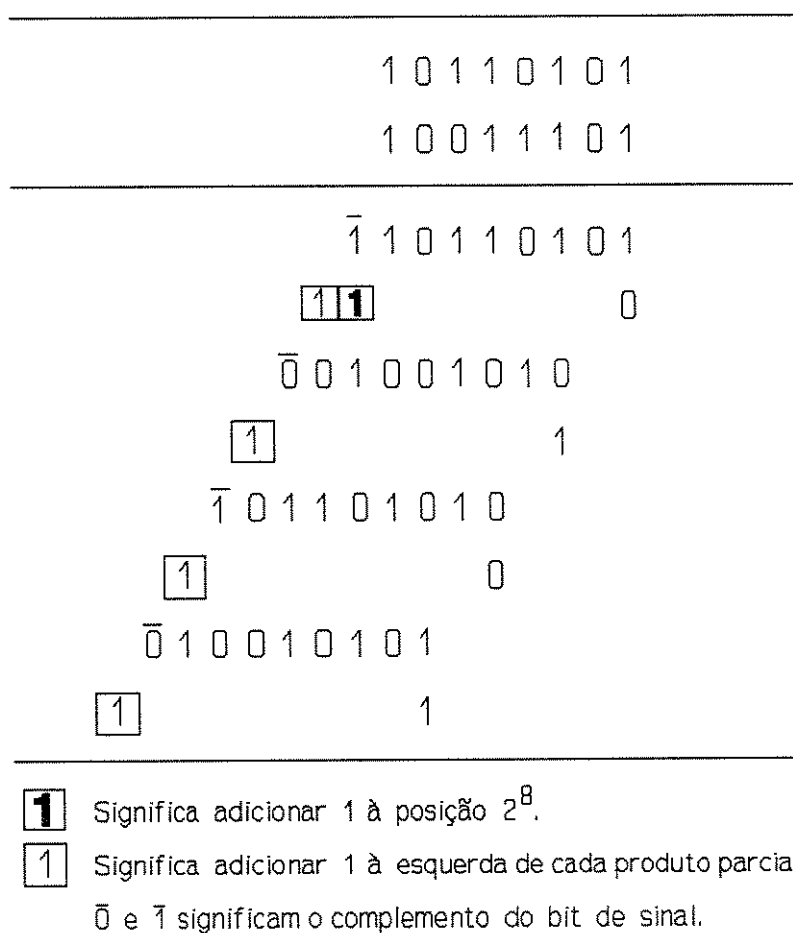


Fig. 3.8 Exemplo do Método do Sinal Gerado

Com as informações disponíveis nesta Seção, resta ainda implementar um bloco que trate do algoritmo de recodificação. O Bloco C3, ilustrado na Fig. 3.9, realiza esta função, recebendo três bits sequenciais do operando Y e gerando os cinco sinais de seleção necessários à recodificação por par de bits.

Um multiplicador de Booth que usa este método está ilustrado em blocos na Fig. 3.10. As três operações que são necessárias para gerar os bits de sinal

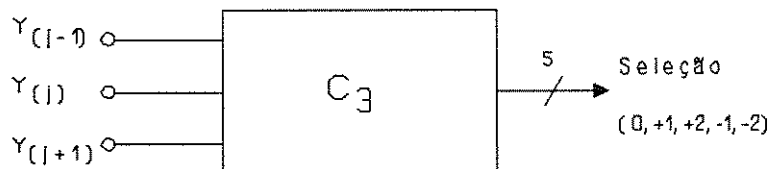


Fig. 3.9 O Bloco C3 do Algoritmo de Recodificação

envolvem, como um primeiro passo, a negação do bit de sinal. Isto pode ser conseguido facilmente trocando-se o operando X pelo seu complemento de um e vice-versa, quando eles dão entrada na lógica C1.

O Cap. 4 tratará da implementação do multiplicador paralelo, ficando o multiplicador série para ser apresentado, de forma mais detalhada, apenas no Cap.5.

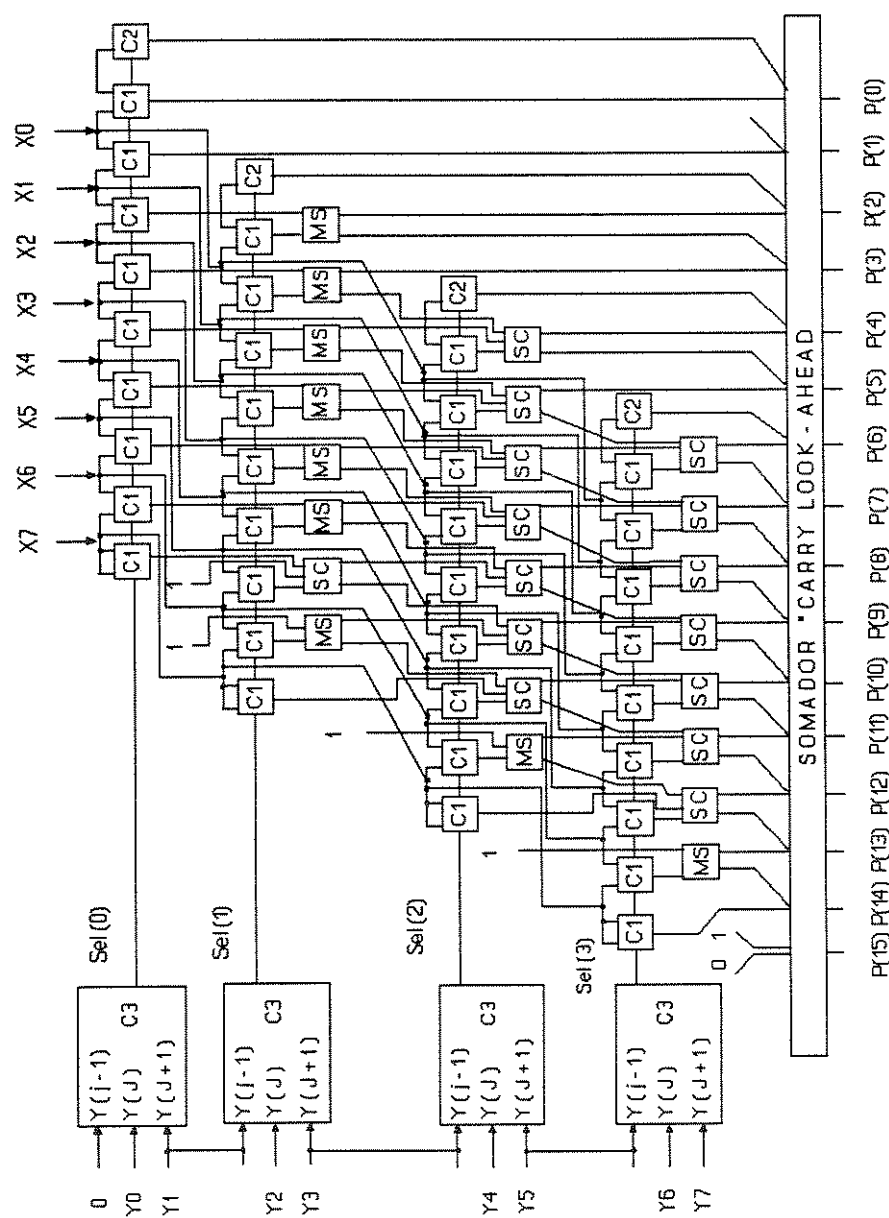


Fig. 3.10 Multiplicador de BOOTH - Método do Sinal Gerado

CAPÍTULO 4

O MULTIPLICADOR PARALELO

Este Capítulo trata do projeto do multiplicador paralelo, difundido no PMU CMOS6 de 2 μm , através da fundição francesa ES2. Embora o Capítulo anterior tenha discutido prioritariamente o multiplicador paralelo de oito bits, considerando o consumo de área do PMU, foi implementado um projeto de multiplicador de apenas quatro bits.

Como comprovação da modularidade da estrutura, que permite o ajuste para qualquer número de bits, apresentar-se-ão também os resultados das simulações de um multiplicador de 32 bits que se situa no limite do algoritmo de Booth utilizado.

Na sequência, será mostrado o diagrama lógico dos circuitos C1, C2, C3

e do somador completo, tratados no Capítulo anterior, e que se aplicam à estrutura do multiplicador também já discutida.

4.1 O CIRCUITO C3

O circuito que trata do algoritmo de recodificação C3 é o responsável pela recodificação por par de bits do operando Y de acordo com a Tab. 3.3. O circuito deve receber três bits sequenciais [Y(j-1), Y(j) e Y(j+1)] do operando Y e entregar as cinco saídas de seleção, C0 a C4, que indicam qual é o dígito recodificado para os bits do operando, ou seja, 0, +1, +2, -1 ou -2. A Tab. 4.1 mostra a lógica a ser implementada para o codificador C3.

As seguintes expressões booleanas podem ser obtidas na tabela:

$$C0 = \bar{Y}(j+1) \cdot \bar{Y}(j) \cdot \bar{Y}(j-1) + Y(j+1) \cdot Y(j) \cdot Y(j-1)$$

$$C1 = \bar{Y}(j+1) \cdot \bar{Y}(j) \cdot Y(j-1) + \bar{Y}(j+1) \cdot Y(j) \cdot \bar{Y}(j-1)$$

$$C2 = \bar{Y}(j+1) \cdot Y(j) \cdot Y(j-1)$$

$$C3 = Y(j+1) \cdot Y(j) \cdot Y(j-1)$$

$$C4 = Y(j+1) \cdot \bar{Y}(j) \cdot Y(j-1) + Y(j+1) \cdot Y(j) \cdot \bar{Y}(j-1)$$

que conduzem ao circuito lógico mostrado na Fig. 4.1.

Dígito Recodificado	Y(j+1)	Y(j)	Y(j-1)	C4	C3	C2	C1	C0
0	0	0	0	0	0	0	0	1
+1	0	0	1	0	0	0	1	0
+1	0	1	0	0	0	0	1	0
+2	0	1	1	0	0	1	0	0
-2	1	0	0	0	1	0	0	0
-1	1	0	1	1	0	0	0	0
-1	1	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0	1

Tabela 4.1 Tabela Verdade para o Codificador C3

4.2 O CIRCUITO C1

Este circuito multiplexador é responsável pelo reposicionamento dos bits do operando X de acordo com as Tabs. 3.6 e 3.7. A Fig. 4.2 ilustra um circuito simples que implementa este circuito no multiplicador.

4.3 O CIRCUITO C2

O C2 é um gerador de soma 1, a ser utilizado nos casos em que o dígito

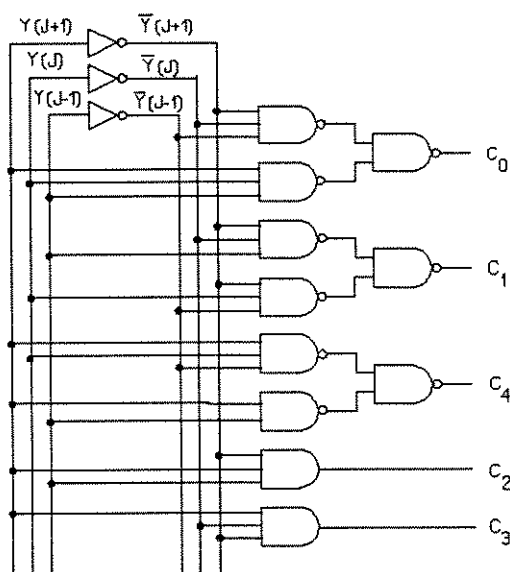


Fig. 4.1 O Circuito Lógico do Codificador C3

recodificado de Booth assume os valores -1 ou -2 e de acordo com as Tabs. 3.6 e 3.7. Para a sua implementação, não há necessidade de se criar um novo circuito, pois basta que se tome o Circuito C1 e que se faça as entradas $X(i)$ e $X(i-1)$ iguais a zero.

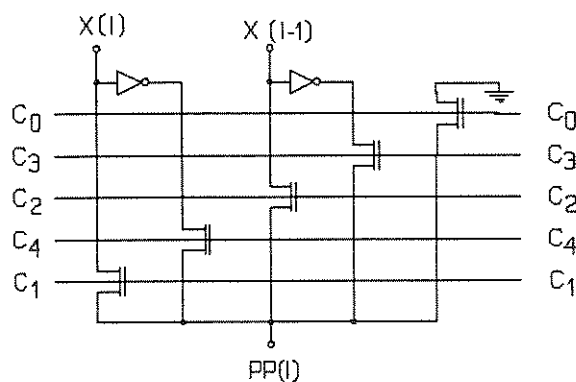


Fig. 4.2 O Multiplexador C1

Colocando de outra maneira, apenas nos casos em que as entradas de seleção C3 ou C4 são ativadas, o que corresponde respectivamente aos dígitos recodificados -1 e -2, PP(i) será feito igual a 1 e poderá ser somado ao bit LSB para se conseguir o complemento de dois do operando.

4.4 OS CIRCUITOS SOMADORES

Na implementação do multiplicador paralelo, discutido nas Seções anteriores, houve a necessidade de se utilizar quatro tipos de somadores, que serão apresentados a seguir.

4.4.1 O CIRCUITO DO MEIO-SOMADOR (MS)

Este circuito, de simples implementação, dispõe de duas entradas e de duas saídas. A sua tabela verdade está mostrada na Fig. 4.3 juntamente com o seu diagrama lógico.

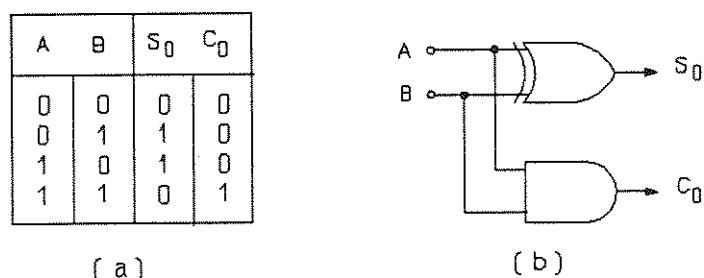


Fig. 4.3 O Meio-Somador

4.4.2 O CIRCUITO DO SOMADOR COMPLETO (SC)

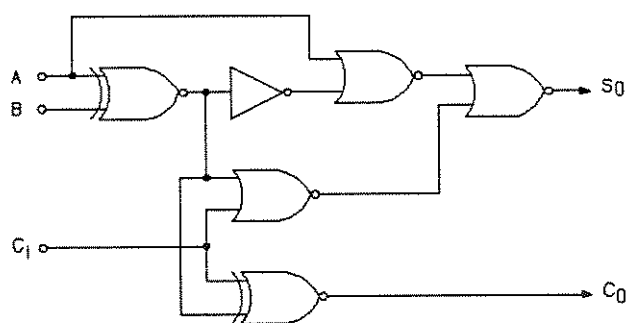
Trata-se de um circuito convencional com três entradas e duas saídas. A Fig. 4.4 mostra a sua tabela verdade e o seu diagrama lógico.

A	B	C_i	S_0	C_0
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$S_0 = C_i \oplus (A \oplus B)$$

$$C_0 = AB + C_i (A \oplus B)$$

(a)



(b)

Fig. 4.4 O Somador Completo

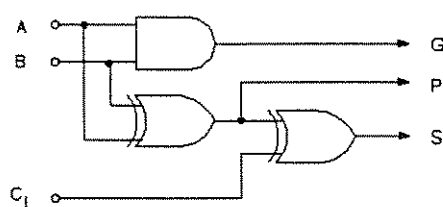
4.4.3 O CIRCUITO DO SOMADOR COMPLETO ESPECIAL (SCE)

Este circuito é um somador especial de três entradas e três saídas adequado à utilização com o circuito do "carry look-ahead (CLA)", a ser discutido posteriormente. A Fig. 4.5 mostra a sua tabela verdade e o seu diagrama lógico.

4.5 O CIRCUITO DO "CARRY LOOK-AHEAD" (CLA)

A	B	C _i	S	P	G
0	0	0	0	0	0
0	0	1	1	0	0
0	1	0	1	1	0
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	0	1	0
1	1	0	0	0	1
1	1	1	1	0	1

(a)



(b)

Fig. 4.5 Somador Especial para o CLA

Este circuito é responsável pela rápida propagação dos "carries" resultantes das somas dos produtos parciais, acelerando a computação. O circuito está mostrado na Fig. 4.6, para o multiplicador paralelo de 4 bits.

4.6 PORTAS LÓGICAS ESPECIAIS

Na implementação do multiplicador paralelo de 4 bits houve ainda a necessidade de se implementar portas especiais denominadas AND4, AND5, AND6, AND7, OR4, OR5, OR6 e OR7, cujos diagramas lógicos estão ilustrados na Fig. 4.7. Prioritariamente, foram desenvolvidas para auxiliarem na implementação do circuito do "carry look-ahead".

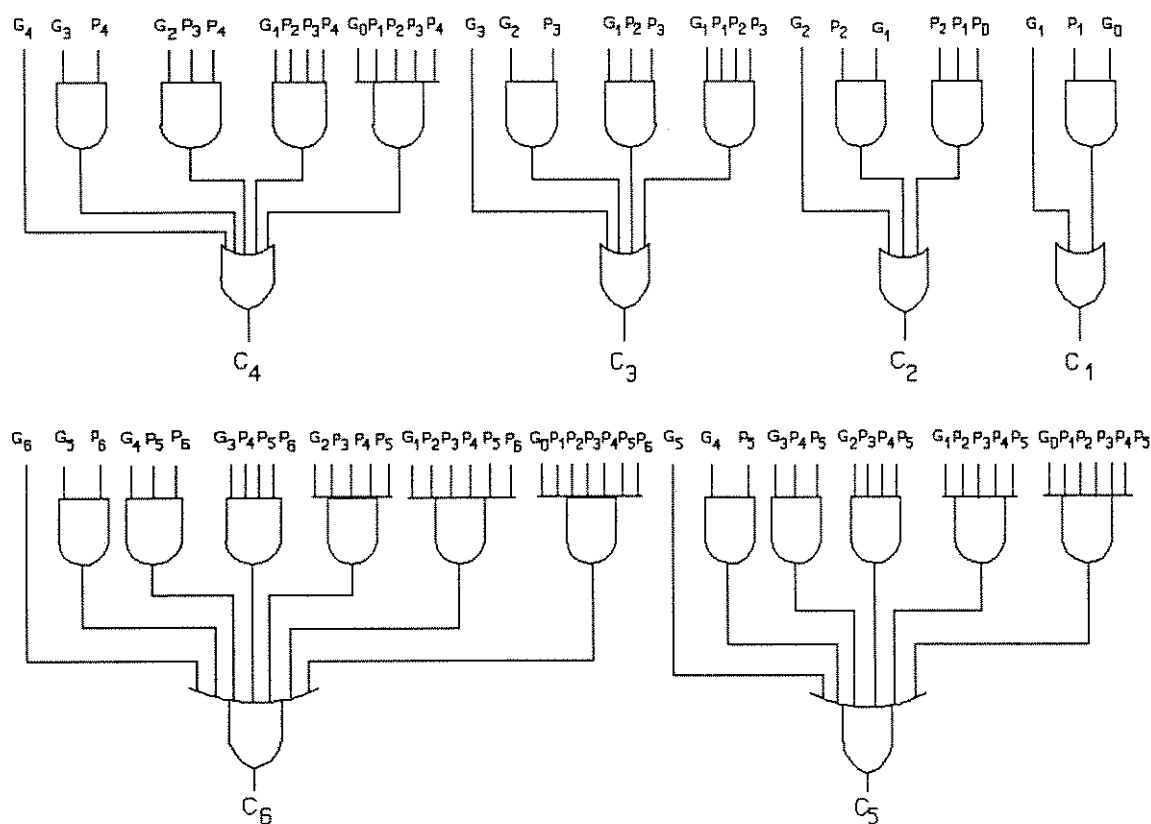


Fig. 4.6 O Circuito "Carry Look-Ahead"

4.7 O DIAGRAMA EM BLOCOS DO MULTIPLICADOR

A Fig. 4.8 traz o diagrama em blocos do multiplicador paralelo de 4 bits onde se pode identificar com facilidade os vários circuitos já apresentados nesta Seção. É fácil perceber-se que, devido a modularidade, multiplicadores com

números maiores de bits podem ser implementados com muita facilidade.

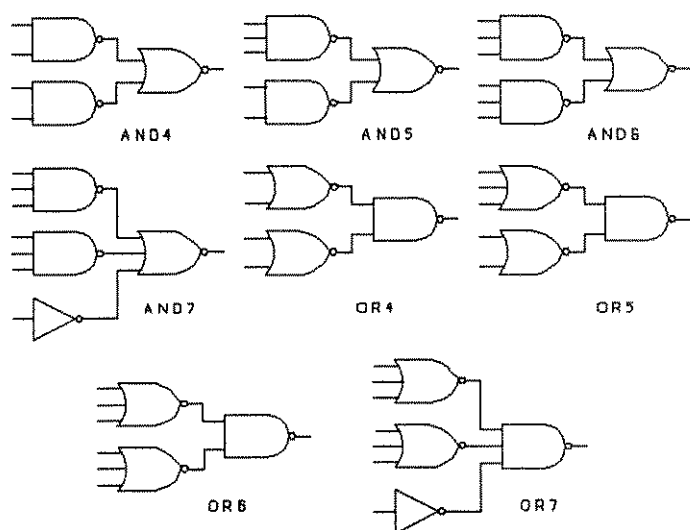


Fig. 4.7 Portas Lógicas Especiais

4.8 AS SIMULAÇÕES

4.8.1 A SIMULAÇÃO ELÉTRICA

Utilizando-se o simulador elétrico "PSPICE Versão 2.G", todos os blocos do multiplicador, mostrados na Fig. 4.8, sofreram simulação elétrica nas seguintes condições:

a) caso típico, caracterizado por modelos de transistores típicos, temperatura de 27°C, tensão de alimentação TTL compatível de 5 Volts;

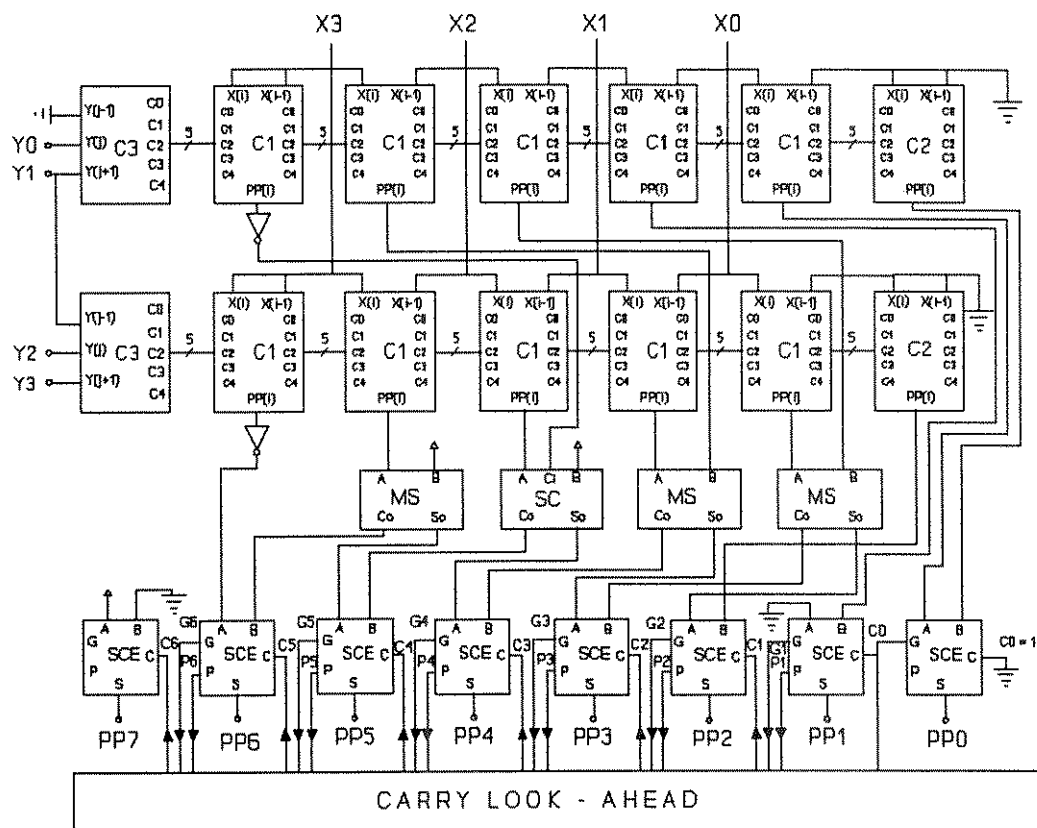


Fig. 4.8 Diagrama em Blocos do Multiplicador Paralelo

- b) pior caso de velocidade com tensão de alimentação de 4.5 Volts, transistores lentos e temperatura de 80°C;
- c) pior caso de potência, caracterizado por uma tensão de alimentação de 5.5 Volts, temperatura de 0°C e transistores rápidos;
- d) pior zero, simulado com transistores tipo n lentos, tipo p rápidos, temperatura de

80°C e tensão de alimentação de 4.5 Volts.

e) pior um, considerado com temperatura de 80°C, tensão de alimentação de 4.5 Volts, transistores p lentos e n rápidos;

f) apenas o sinal situado no caminho crítico da temporização foi considerado.

As simulações conduziram a um grande volume de informações, impossibilitando a anexação a este trabalho. Entretanto, encontram-se disponíveis aos interessados, uma vez que apenas os resultados sintetizados serão apresentados na sequência.

Com base na Fig. 4.8, pode-se perceber que, quando o número de bits do multiplicador cresce, apenas os blocos do Codificador C3 e do circuito do "carry look-ahead" terão os seus tempos alterados. Por isso, serão apresentados os resultados de simulação para multiplicadores de 4 bits e de 32 bits, faixa em que o Algoritmo de Booth exibe a sua melhor performance.

O BLOCO DO CODIFICADOR C3

A simulação deste bloco, permitiu detetar que o caminho crítico do sinal, em termos de propagação, é estabelecido através do sinal C4 do codificador. No pior caso de velocidade, foram obtidos os tempos indicados na Tab. 4.2, nas transições de saída de baixo para alto e de alto para baixo e para um multiplicador de 4 bits. A Tab. 4.3 mostra os mesmos tempos para o sinal C4, que representa o caminho crítico no

multiplicador de 32 bits.

CASOS	Sinal C4 - Tempo de Propagação (nS)	
	Baixo para Alto	Alto para Baixo
TÍPICO	2.4	3.4
PIOR CASO DE VELOCIDADE	5.2	6.6
PIOR CASO DE POTÊNCIA	1.8	2.4

Tabela 4.2 Temporização do Codificador C3 - 4 Bits

CASOS	Sinal C4 - Tempo de Propagação (nS)	
	Baixo para Alto	Alto para Baixo
TÍPICO	9.7	21.0
PIOR CASO DE VELOCIDADE	23.2	32.1
PIOR CASO DE POTÊNCIA	8.8	13.4

Tabela 4.3 Temporização do Codificador C3 - 32 Bits

O BLOCO DO MULTIPLEXADOR C1

A simulação deste bloco conduziu aos tempos mostrados na Tab. 4.4,

onde se pode perceber que o tempo de propagação mais crítico corresponde a um valor de 0.9 nS na transição de alto para baixo na condição de pior caso de velocidade.

CASOS	Sinal PP(i) - Tempo de Propagação (nS)	
	Baixo para Alto	Alto para Baixo
TÍPICO	0.5	0.5
PIOR CASO DE VELOCIDADE	0.5	0.9
PIOR CASO DE POTÊNCIA	0.4	0.4

Tabela 4.4 Temporização do Multiplexador C1

Conforme já comentado anteriormente, o Bloco C2 é um caso particular do C1, onde as entradas são feitas igual a zero. Logo, os tempos mostrados na Tab.4.4 também se aplicam ao Bloco C2.

OS BLOCOS DOS SOMADORES

As Tabs. 4.5, 4.6 e 4.7 resumem, respectivamente, os tempos de propagação determinados pela simulação elétrica do caminho crítico dos circuitos meio-somador, somador completo e somador completo especial, já discutidos anteriormente, no caso típico, pior caso de velocidade e pior caso de potência, vitais na análise de performance do multiplicador.

CASOS	Sinal S_0 - Tempo de Propagação (nS)	
	Baixo para Alto	Alto para Baixo
TÍPICO	0.7	0.7
PIOR CASO DE VELOCIDADE	0.9	0.9
PIOR CASO DE POTÊNCIA	0.9	0.9

Tabela 4.5 Temporização do Meio Somador

CASOS	Sinal C_0 - Tempo de Propagação (nS)	
	Baixo para Alto	Alto para Baixo
TÍPICO	1.3	1.0
PIOR CASO DE VELOCIDADE	1.5	1.2
PIOR CASO DE POTÊNCIA	0.5	0.4

Tabela 4.6 Temporização do Somador Completo

No caso do somador completo especial, foi adicionada, para efeito de simulação, uma capacitância equivalente a dez cargas CMOS na saída S deste somador, que coincide com a saída do multiplicador paralelo. Este é o motivo pelo qual seu atraso de propagação ser tão significativo.

CASOS	Sinal S - Tempo de Propagação (nS)	
	Baixo para Alto	Alto para Baixo
TÍPICO	4.9	4.6
PIOR CASO DE VELOCIDADE	8.9	8.5
PIOR CASO DE POTÊNCIA	3.1	2.8

Tabela 4.7 Temporização do Somador Especial

O BLOCO DO "CARRY LOOK-AHEAD"

É fácil perceber-se, com base na Fig. 4.6, que o circuito gerador do "carry 6" deste bloco é o que conduz aos maiores atrasos de propagação no multiplicador paralelo de 4 bits. Assim, após a simulação elétrica do circuito, chegou-se aos resultados que estão mostrados na Tab. 4.8.

CASOS	Sinal C6 - Tempo de Propagação (nS)	
	Baixo para Alto	Alto para Baixo
TÍPICO	2.8	2.8
PIOR CASO DE VELOCIDADE	5.1	5.3
PIOR CASO DE POTÊNCIA	1.5	1.6

Tabela 4.8 Temporização do "Carry 6"

Adicionalmente, simulou-se o caminho crítico do circuito do "carry 62" para o multiplicador paralelo de 32 bits. Os resultados estão resumidos na Tab. 4.9.

CASOS	Sinal C62 - Tempo de Propagação (nS)	
	Baixo para Alto	Alto para Baixo
TÍPICO	4.6	5.5
PIOR CASO DE VELOCIDADE	9.3	10.4
PIOR CASO DE POTÊNCIA	2.6	3.0

Tabela 4.9 Temporização do "Carry 62"

4.8.2 ANÁLISE DA FREQUÊNCIA DE OPERAÇÃO

Com base nas Tabs. 4.2 a 4.9 e na Fig. 4.8, procedeu-se à uma análise dos tempos de propagação dos multiplicadores de 4 e de 32 bits.

No caso do multiplicador de 4 bits, a soma dos tempos de propagação dos blocos conduziu a um resultado em pior caso de 23,7 nS, que corresponde a uma frequência de operação máxima de 42 MHz. Entretanto, considerando os atrasos proporcionados pelos "pads" de entrada e de saída, a frequência de operação deverá se situar em torno de 35 MHz.

Com base nos resultados das simulações, é possível concluir-se que esta

frequência pode ser melhorada aumentando-se as dimensões dos transistores utilizados no codificador C3. Entretanto, o preço a ser pago é o aumento da área de integração. Um interessante estudo poderia ser feito na busca do melhor compromisso entre velocidade e área; entretanto, tal estudo não foi objeto deste trabalho onde se optou pela solução que conduziu à menor área de integração.

A análise anterior foi realizada também para o multiplicador de 32 bits, chegando-se a um tempo de atraso total em pior caso de 54,3 nS, que corresponde a uma frequência máxima de operação de 18,5 MHz. Com a inserção dos "pads", esta frequência é reduzida para algo em torno de 15 MHz.

Pode-se perceber, neste caso, a excelente performance do circuito "carry look-ahead" que permite apenas um pequeno aumento do tempo de propagação quando se passa do multiplicador de 4 para o de 32 bits. Entretanto, a complexidade do hardware e o aumento da área integrada crescem numa proporção muito elevada.

Adicionalmente, podem-se aplicar ao multiplicador de 32 bits as mesmas considerações apresentadas acima para o de 4 bits, em relação à performance do codificador C3 (vide Fig. 4.8).

Outra consideração importante a ser feita refere-se ao circuito do "carry look-ahead" no multiplicador de 32 bits. As dimensões deste circuito atingem proporções que poderão comprometer a implementação física, tornando obrigatória a proposição de outras lógicas para a sua execução, como a utilização de uma solução que empregue módulos de "carry look-ahead" de 4 bits, conectados serialmente. Seguramente, esta solução, como qualquer outra proposta, implicará em redução da frequência de operação do multiplicador.

4.8.3 A SIMULAÇÃO LÓGICA

Embora os resultados da simulação elétrica já apontassem para um projeto logicamente correto, procedeu-se à simulação lógica de todos os blocos do multiplicador, tratados com base na Fig. 4.8, utilizando-se o Simulador Hilo 3M instalado em um computador VAX. O volume dos resultados obtidos impede a sua anexação a este trabalho, ficando à disposição de possíveis interessados.

4.9 A ELABORAÇÃO DO "LAYOUT" DO MULTIPLICADOR

Para a edição do "layout" do multiplicador paralelo de 4 bits, conseguiu-se no PMU CMOS6 em tecnologia de dupla camada de metal da ES2, uma área livre de apenas 4 mm por 700 μm .

Inicialmente, pretendia-se incluir, de forma separada, todos os blocos apresentados na Fig. 4.8, assegurando com sucesso a realização dos testes durante a fase de caracterização do multiplicador. Entretanto, uma posterior análise mostraria a impossibilidade de tal implementação na área disponível.

Considerando-se a inexperiência e o fato de que o multiplicador ocupava uma área de 3.924 μm por 659 μm (incluindo os seus dezoito pads), decidiu-se tentar implementá-lo totalmente sem criar mecanismos que facilitassem os seus testes. Já se sabia que haveria dificuldades, que poderiam ser minimizadas, utilizando-se a estrutura do Laboratório de Testes do CTI em Campinas.

O "LAYOUT"

Com a preocupação de se minimizar áreas integradas, todos os transistores MOS de canal n foram implementados com as dimensões de $L = 2\mu\text{m}$ e $W = 3\mu\text{m}$, situadas no limite das regras de projeto da ES2. Já os transistores MOS de canal p contaram com $L = 2\mu\text{m}$ e $W = 6\mu\text{m}$ para que a mobilidade no canal fosse compensada.

Utilizando-se o editor gráfico KIC, o "layout" das várias células do multiplicador paralelo foram implementadas. A montagem final foi realizada no CTI, onde o "layout" final foi submetido ao DRC (Design Rules Checking), com sucesso, adquirindo a configuração mostrada na Fig. 4.9, gerada em um "hardcopy". Infelizmente, o CTI não dispunha de um software extrator de parâmetros, elevando seriamente a possibilidade de ocorrência de conexões erradas.

4.10 RESULTADOS E CONCLUSÕES

A Fig. 4.10 ilustra a fotomicrografia do multiplicador paralelo, onde se pode perceber a inteira ocupação da área disponível no PMU, impossibilitando a inclusão de estruturas de testes.

Como já comentado anteriormente, também por uma questão de área, foi implementada uma estrutura de 4x4 bits. Entretanto, as simulações indicaram que uma estrutura de 32x32 bits e, portanto, próxima ao limite do algoritmo utilizado, poderia ser utilizada com amplo sucesso.

As simulações indicaram ainda que a estrutura paralela de 4x4 bits podia trabalhar com taxas de multiplicação de até 35 MHz, o que representa um excelente desempenho para multiplicadores desta natureza.



Fig. 4.10 Microfotografia do Multiplicador Paralelo

No caso da estrutura de 32x32 bits a frequência cai para 15 MHz, o que é surpreendente pois acreditava-se que esta estrutura seria muito mais lenta. A pequena queda ocorrida deve-se, principalmente, ao bom comportamento do somador "carry

look-ahead" agregado ao projeto.

Os primeiros testes de bancada indicaram que problemas de "layout" impediriam o funcionamento do multiplicador desta primeira difusão. A não existência de um laboratório equipado para este tipo de diagnóstico, fez com que se optasse por decapar quimicamente os "chips", na tentativa de se acessar as trilhas do circuito.

O ataque químico usualmente altera o comportamento elétrico dos componentes e nem sempre assegura sucesso no contato às trilhas, fato verificado durante esta fase dos trabalhos.

A impossibilidade de se acessar o circuito impediu que qualquer resultado prático e confiável pudesse ser aqui anexado.

CAPÍTULO 5

A IMPLEMENTAÇÃO DO MULTIPLICADOR SÉRIE

A maioria dos sistemas de processamento de dados, tais como os filtros digitais, usa representações binárias série para sequenciar os seus dados. Mesmo alguns computadores de propósito geral empregam a aritmética série em sua arquitetura. Entretanto, o maior problema associado a esta aritmética tem sido o tempo ou a complexidade do circuito necessária à realização da multiplicação.

Buscou-se, neste projeto, uma estrutura que não conduzisse a um circuito de alta complexidade, onde fosse preservada a modularidade e que pudesse operar em velocidades relativamente altas.

5.1 CONSIDERAÇÕES GERAIS

A necessidade de se realizar multiplicações de forma bem rápida levou ao desenvolvimento de um multiplicador "pipeline" em 1.968 [14]. Embora tivesse uma estrutura modular, o multiplicador aceitava apenas operandos positivos.

A representação de números negativos na forma de complemento de dois, tão conveniente para a adição e subtração, representava um problema para a multiplicação. FREENY et al [15] apresentaram uma solução para o problema com uma estrutura dividida que infelizmente trazia a desvantagem de se perder a modularidade.

O inconveniente acima levou SPINKS [16] a construir um multiplicador série "pipeline" que usava diretamente operandos em complemento de dois. Seu circuito usava um multiplicador "pipeline" simples seguido por um somador que adicionava um termo de correção função do multiplicador e do sinal do multiplicando.

LYON [17] mostrou que uma simples modificação do multiplicador "pipeline" modular permitia que se executassem adições por complemento de dois de produtos parciais, tal que operandos em complemento de dois podiam ser multiplicados por coeficientes positivos sem que se empregasse um termo de correção.

A partir deste multiplicador, o circuito podia ser modificado levemente pela adição de um último estágio, em lugar de outros módulos, para manipular tanto sinal e magnitude ou coeficientes em complemento de dois.

Alternativamente, uma modularidade plena podia ser conseguida usando-se um multiplicador recodificado, como no Algoritmo de Booth, que foi a

opção definida neste projeto, pois além da estrutura "pipeline" em complemento de dois, o algoritmo permite uma desejável aceleração na computação.

5.2 O MULTIPLICADOR RECODIFICADO

O Algoritmo de Booth pode ser usado para se construir um multiplicador série em complemento de dois em que todos bits do multiplicador são tratados igualmente. O método, já discutido, é equivalente a se recodificar o multiplicador como uma sequência de dígitos ternários $(-1, 0, +1)$ com peso binário.

O multiplicador original,

$$Y = \sum_{i=1}^{k-1} Y_i 2^{i-k+1} - 2Y_k \quad (5-1)$$

pode ser reescrito como [2],[9] :

$$Y = \sum_{i=1}^k Y'_i 2^{i-k+1} \quad (5-2)$$

onde,

$$Y'_i = Y_{i-1} - Y_i \in \{-1, 0, +1\}$$

e Y_0 é tomado como zero.

A multiplicação é realizada somando-se os produtos parciais nulos, negativos ou positivos obtidos pela multiplicação de cada Y_i pelo multiplicando. A adição precisa ser realizada em complemento de dois usando-se a extensão do sinal, quando necessário.

A Fig. 5.1 mostra um módulo possível de ser utilizado, proposto por LYON [17] em que, como solução para se controlar a soma/subtração, optou-se pela aproximação de se implementar um complementador na entrada Y do somador juntamente com o circuito de controle "clear/preset" do "carry". Outros módulos, utilizando outras aproximações, podem ser facilmente implementados.

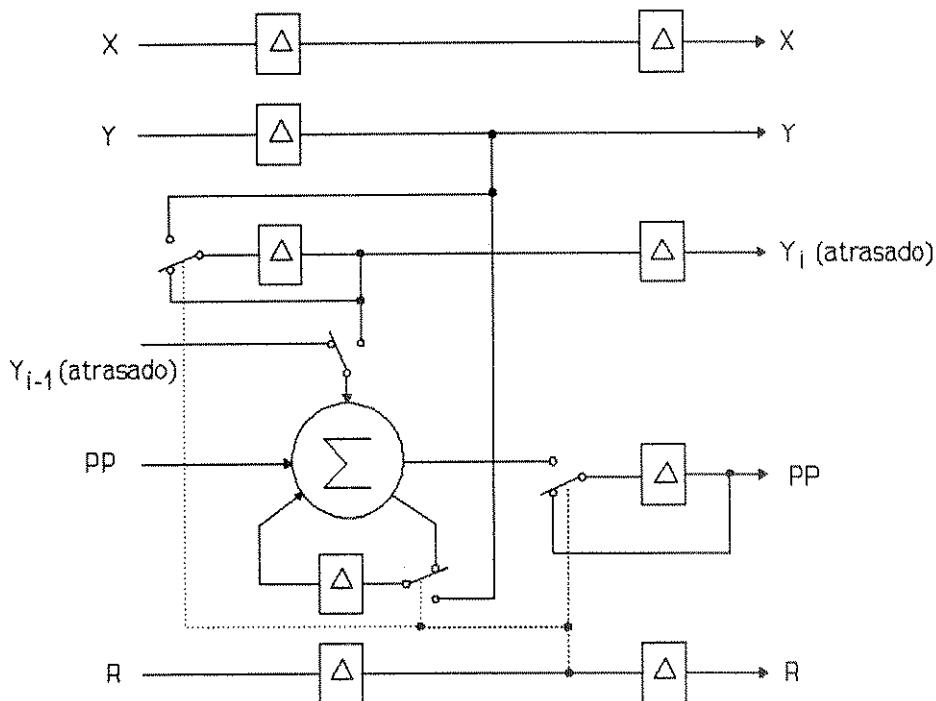


Fig. 5.1 Módulo de Multiplicação Série Utilizando Booth

5.2.1 RECODIFICAÇÃO UTILIZANDO O ALGORITMO DE BOOTH MODIFICADO

O multiplicador pode também ser recodificado como:

$$Y = \sum_{i=1}^k Y_i' 2^{i-k+1} \text{ para } i=\text{ímpar} \quad (5-3)$$

onde,

$$Y_i' \in \{-2, -1, 0, +1, +2\}$$

A multiplicação utilizando esta representação, que também foi utilizada na implementação do multiplicador paralelo, requer somente $K/2$ somadores / subtratores e circuitos de controle para se escolher X , $2X$ ou zero, como será discutido posteriormente.

Uma estrutura "pipeline" bastante interessante para um módulo de multiplicação, utilizando o Algoritmo de Booth Modificado, foi proposta por E.K.CHENG em julho de 74 no California Institute of Technology (trabalho não publicado) e apresentada em [17]. A Fig. 5.2 mostra esta célula que foi escolhida como básica para a implementação do multiplicador série deste trabalho.

Cada célula do multiplicador contém alguns elementos de atraso a mais que os de outras lógicas encontradas na literatura, mas somente uma célula é necessária para cada dois bits do coeficiente.

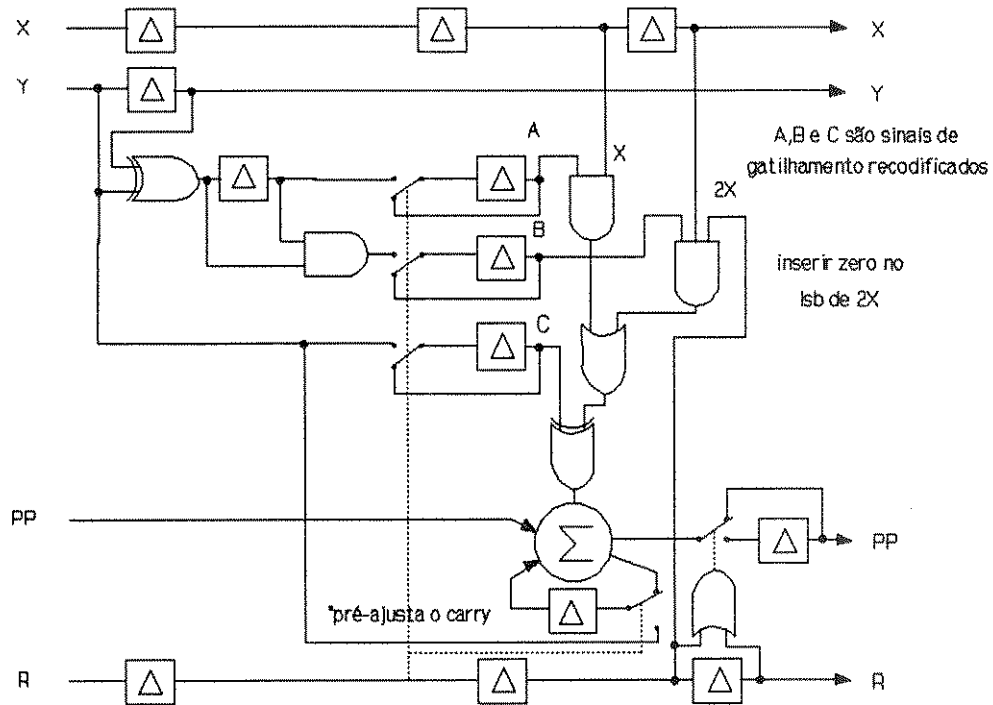


Fig. 5.2 A Célula Básica de Multiplicação de Cheng

Algumas vezes pode ser vantajoso colocar a lógica de recodificação em um bloco antes dos módulos e enviar três bits a cada estágio via dois registradores em lugar do simples registrador de coeficiente. Isto economiza lógica mas destrói a modularidade, pelo que não será utilizado.

5.3 O PROJETO DO MULTIPLICADOR

O projeto do multiplicador série, que será tratado neste Capítulo, foi difundido no PMU CMOS6 de $2\ \mu\text{m}$, através da fundição francesa ES2. A estrutura

proposta, como já discutido, foi a de um multiplicador "pipeline" utilizando o Algoritmo de Booth Modificado com base na Célula de CHENG, já apresentada na Seção anterior.

Por considerações de economia de área, realizou-se o projeto de um multiplicador de 8 bits, embora a estrutura possa ser ajustada facilmente a qualquer número de bits, face a sua modularidade.

A Fig. 5.3 mostra o diagrama em blocos do circuito utilizado no projeto do multiplicador série. Os operandos $X = X_7X_6X_5X_4X_3X_2X_1X_0$ e $Y = Y_7Y_6Y_5Y_4Y_3Y_2Y_1Y_0$ são carregados simultaneamente nos seus respectivos registradores de deslocamento paralelo-série "RDPS", que também recebem os sinais de controle CRPS e PRPS e o sinal de "clock" externo CK2, tratados posteriormente.

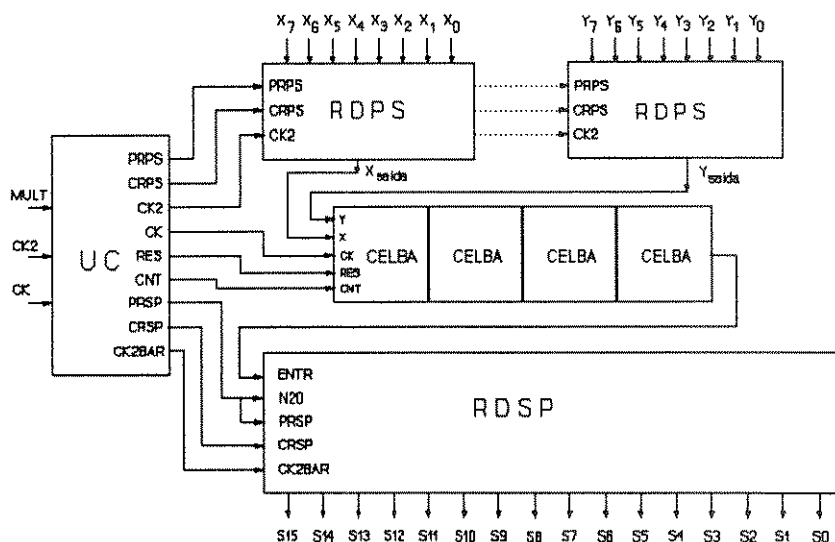


Fig. 5.3 Diagrama em Blocos do Multiplicador Série de 8 Bits

Os operandos X e Y, de forma sincronizada, são deslocados serialmente para as células básicas de multiplicação CELBA, que executam a multiplicação após terem realizado a recodificação por par de bits do Algoritmo de Booth Modificado. O resultado é enviado para o registrador de deslocamento série-paralelo "RDSP", que compõe a palavra de 16 bits e a deixa apta a ser lida externamente, de forma paralela.

Finalizando, a Unidade de Controle recebe os sinais de clock externos CK e CK2, bem como o sinal de habilitação para iniciar a multiplicação gerando, por sua vez, os sinais PRPS, CRPS, RES, CNT, PRSP e CRSP, que sincronizam os demais blocos, como discutido posteriormente.

As próximas Seções apresentarão em detalhes os vários blocos que compõem o multiplicador série proposto (Fig. 5.3).

5.3.1 A CÉLULA BÁSICA CELBA

A célula básica de multiplicação, utilizada neste projeto, está mostrada na Fig. 5.4. Pode-se perceber que, estruturalmente, assemelha-se à proposta por CHENG.

O funcionamento da estrutura pode ser melhor entendido, considerando-se uma multiplicação de operandos com apenas quatro bits. O número de células básicas necessárias é de $N/2$, se N for par e de $(N+1)/2$, se N for ímpar, onde N é o número de bits do multiplicador. Neste exemplo, serão necessárias apenas duas células.

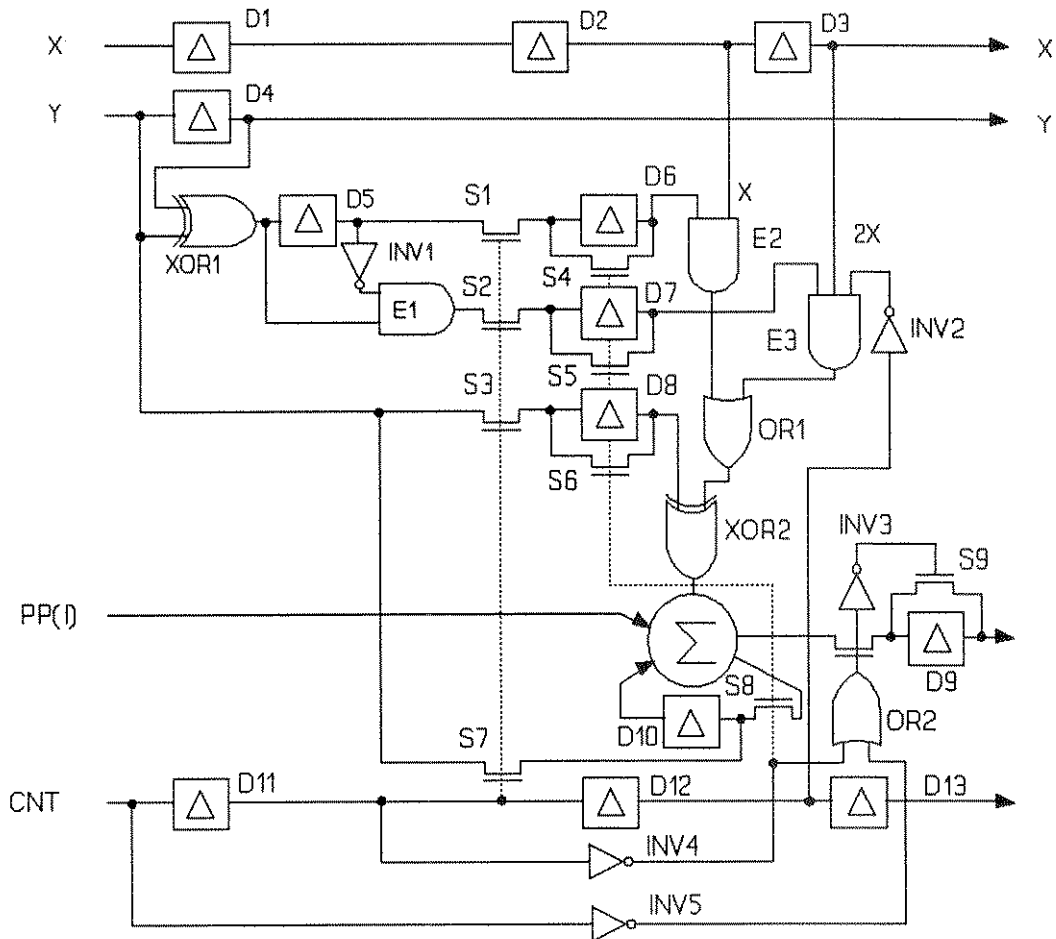


Fig. 5.4 A Célula Básica de Multiplicação CELBA

Considere-se inicialmente que a célula básica da Fig. 5.4 irá receber os dois bits menos significativos do multiplicador Y feitos neste exemplo como 1 0 1 1 , ou seja:

$$\begin{array}{cc} Y_1 & Y_0 \\ 1 & 0 \end{array} \mid \begin{array}{cc} Y_3 & Y_2 \\ 1 & 1 \end{array}$$

que corresponde ao dígito -1 do algoritmo de Booth.

OS BLOCOS D4, D5, XOR1, INV1 e E1

Os blocos lógicos XOR1, INV1, E1 e os atrasos D4 e D5, são responsáveis pela recodificação do multiplicador Y . Assim, a primeira célula básica deve analisar $Y_1Y_0 = 110$. O 0 à direita do bit menos significativo é proveniente do Algoritmo de Booth. Por este motivo, é necessário, na inicialização de cada multiplicação, zerar as saídas de todos os atrasos mostrados na Fig. 5.4, que na realidade correspondem a flip-flops D.

Do exposto, no primeiro período de clock após um "reset" geral, ter-se-á um 0 na saída do bloco D4 e na sua entrada o bit $Y_0 = 1$. Estes dois sinais também dão entrada no bloco XOR1, resultando em um sinal "1" na sua saída. Durante este período, os demais sinais são irrelevantes.

No segundo período de clock, o sinal "1", que se encontrava na entrada do bloco D4, é levado à sua saída e o bit $Y_1 = 1$ é aplicado em sua entrada. Simultaneamente, o sinal "1" da saída do bloco XOR1 passa para a saída de D5. Na sequência, o bloco XOR1 irá executar um ou-exclusivo de $1 \oplus 1$, que resulta em um zero na sua saída. Como na saída do bloco D5 tem-se "1", a saída do bloco INV1 será "0" e, logo, na saída do bloco E1 ter-se-á $0.0 = 0$.

Ainda durante este segundo período, as chaves S1, S2 e S3 estão ligadas, permitindo que os seus respectivos sinais passem para as entradas dos blocos de atraso D6, D7 e D8.

Após o terceiro período de clock, os sinais gatilhados em D6, D7 e D8 irão para as suas saídas. Estes sinais, gerados de acordo com a recodificação de Booth,

irão autorizar a execução da lógica restante. No caso do exemplo em apresentação, os blocos D6, D7 e D8 estarão retendo, respectivamente, os sinais 1, 0 e 1.

OS BLOCOS E2, E3, OR1, XOR2 E INV2

Estes blocos têm como função permitir a passagem de 0, +1, +2, -2 ou -1, de acordo com o Algoritmo de Booth Modificado, para a entrada do bloco somador. Para tanto, executam a lógica necessária com os sinais disponíveis na saída dos blocos D6, D7 e D8 , após o terceiro período de clock.

Deve-se observar que neste terceiro período, se o sinal na entrada X for $X_3X_2X_1X_0$, ter-se-á X_1 na saída do bloco D1, X_0 na saída do bloco D2 e "0" na saída do bloco D3 (devido ao "reset" inicial).

Como se tem "1" na saída do bloco D6, a saída do bloco E2 será X_0 . Por outro lado, por se ter 0 na saída do bloco D7, a saída de E3 também será nula. Assim, o bloco OR1 executará uma função OU entre 0 e X_0 , que resultará apenas em X_0 .

Finalizando, o bloco D8 tem 1 na sua saída. Assim, o bloco XOR2 executará o OU-Exclusivo $1 \oplus X_0$, que é equivalente a $\bar{X}_0$. Este é o valor que será levado ao somador e que está de acordo com o algoritmo de Booth.

OS BLOCOS D10, S8 E S7

Na multiplicação , resta ainda saber se deve ou não ser adicionado um sinal 1 ao bit menos significativo, que no caso é X_0 . O bloco D10 e as chaves S7 e S8 devem obedecer a uma lógica que permita a execução correta desta decisão.

Analisando o exemplo em questão, tem-se que, após o segundo período de clock, o bit $X_1 = 1$ do sinal da entrada X é levado ao bloco D10, através da chave S7, que se encontra aberta neste período. Porém, como o bloco D10 é um flip-flop D, este sinal somente estará disponível para o "carry" de entrada C_i do somador no terceiro período de clock.

Assim, no terceiro período de clock, o somador executará $A+B+C_i = S+C_0$, que no exemplo corresponde a $0+\bar{X}_0+1 = S+C_{out}$ e que se encontra de acordo com o esperado.

Uma consideração importante a ser feita é que os sinais nas saídas dos blocos D6, D7 e D8 devem ficar constantes até que todos os bits do multiplicando X sejam resolvidos. Em outras palavras, estes sinais devem permanecer constantes no mínimo $N+1$ bits, se N for par e $N+2$ bits, se N for ímpar. No caso do circuito projetado, foi ajustado uma temporização de $N+2$ bits para N par e $N+3$ bits para N ímpar, por razões que serão explicadas posteriormente.

OS BLOCOS INV3, S10, D9 E OR2

A função destes blocos é a de liberar ou não o sinal de saída do bloco somador truncando o resultado de acordo com o algoritmo de Booth. A conexão foi realizada com o intuito de se fazer a abertura da chave S9 simultaneamente com o fechamento das chaves S1, S2 e S3, para assegurar que não haja alteração no dado que deve entrar no somador e, portanto, no dado que deve sair do mesmo.

A chave S9 tem que ficar ligada o tempo necessário para que todos os bits passem para a saída do circuito. Estes dois últimos itens comentados são

a razão da mudança do número de bits em que as saídas dos blocos D6, D7 e D8 têm que ficar constantes, conforme comentado acima.

BLOCOS D11,D12 E D13

A função destes blocos é somente a de acrescentar atrasos em um sinal de controle que é aplicado na entrada do bloco D11 e gerado pela Unidade de Controle do multiplicador. O momento não é oportuno para a sua discussão, que será feita posteriormente.

BLOCO INV2

A função do sinal que vem do bloco INV2 é a de acrescentar 0 ao bit menos significativo X_{-1} , quando o bloco E3 (2X) estiver sendo ativado em uma operação de multiplicação e, portanto, de acordo com o Algoritmo de Booth.

Na sequência, o bloco XOR2 decidirá se este bit continua sendo 0 (que corresponde a +2), ou se será 1 (que corresponde a -2). No terceiro período, onde se começam a resolver os bits do multiplicando, o sinal de saída do bloco INV2 é 0, o que resulta em 0 na saída do bloco E3, independentemente de suas outras entradas. A partir do 4^o. bit, o sinal na saída do bloco INV2 será sempre 1, o que faz com que a saída do bloco E3 seja agora decidida pelas suas duas outras entradas.

É importante observar, também, que a validação do sinal obedece a Tab. 5.1 para multiplicadores de 2 a 16 bits.

As células básicas, além dos operandos X e Y, recebem os sinais CK, RES

e CNT da Unidade de Controle. O sinal de "clock" CK é utilizado na sincronização dos atrasos, o sinal RES permite a zeragem de inicialização da multiplicação e o sinal CNT será discutido na seção que trata da Unidade de Controle.

MULTIPLICADOR DE	SINAL VÁLIDO A PARTIR DO
2 bits	4 ^o bit
3 e 4 bits	5 ^o bit
5 e 6 bits	6 ^o bit
7 e 8 bits	7 ^o bit
9 e 10 bits	8 ^o bit
11 e 12 bits	9 ^o bit
13 e 14 bits	10 ^o bit
15 e 16 bits	11 ^o bit

Tabela 5.1 Validade do Sinal de Saída

O SOMADOR COMPLETO

Na célula básica da Fig. 5.4 está mostrado um somador completo. A sua arquitetura é diferente da utilizada no projeto do multiplicador paralelo. A Fig. 5.5 apresenta o seu diagrama lógico.

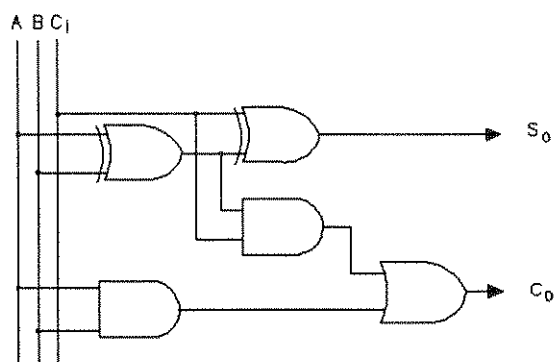


Fig. 5.5 O Somador Completo

OS FLIP-FLOPS D

Para a sincronização das operações internas das células básicas, são utilizados flip-flops do tipo D, representados por atrasos na Fig. 5.4. A Fig. 5.6 mostra o diagrama lógico destes componentes.

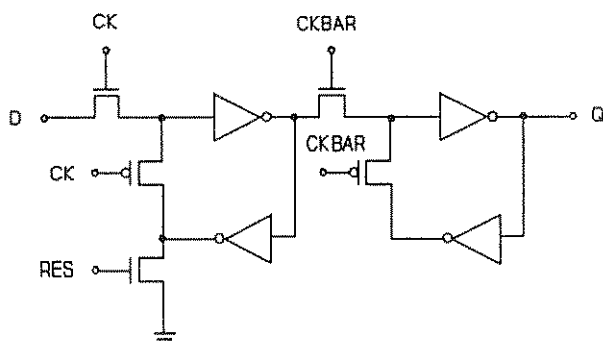


Fig. 5.6 Flip-Flop Tipo D da Célula Básica

5.3.2 OS REGISTRADORES DE DESLOCAMENTO

Uma configuração básica de registrador de deslocamento está mostrada na Fig. 5.7 onde, como ilustração, mostrou-se uma estrutura de quatro bits. A configuração foi utilizada como base para a implementação dos registradores do multiplicador.

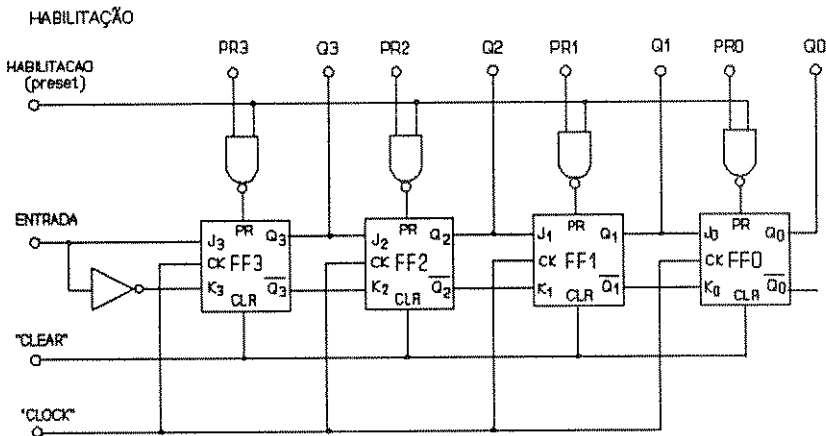


Fig. 5.7 Registrador de Deslocamento de Quatro Bits

Os flip-flops são do tipo JK mestre-escravo com possibilidade de carregamento paralelo através das linhas de "preset". O estágio que armazena o bit mais significativo é convertido em um "latch" tipo D para permitir a entrada serial dos dados.

O REGISTRADOR DE DESLOCAMENTO SÉRIE-PARALELO RDSP

O registrador da Fig. 5.7, para operar como série-paralelo, deve ter inicialmente os seus flip-flops "zerados". Para tanto, basta que se aplique 0 na entrada "clear" e que se mantenha a linha de habilitação do "preset" em nível baixo. Na sequência, a entrada "clear" deve ser levada para 1 quando então o dado série e o "clock" de sincronismo devem ser aplicados.

O bit menos significativo entra no "latch" mestre do flip-flop 3 (FF3), quando o sinal de "clock" CK muda de 0 para 1 pela ação do flip-flop operando como D. Na queda do "clock", o sinal 1 é transferido para o "latch" escravo do FF3 e $Q_3 = 1$, enquanto que todas as outras saídas permanecem inalteradas. Os demais bits da palavra de dado entram serialmente no registrador sincronizados pelo "clock".

No caso do projeto do multiplicador serial, optou-se por um registrador de deslocamento semelhante ao ilustrado, exceto que a sua zeragem é conseguida fazendo-se a linha de "clear" igual a 0 e a de "preset" igual a 1. A Fig. 5.8 mostra a estrutura escolhida onde, por serem desnecessários, eliminaram-se os blocos lógicos Não-E.

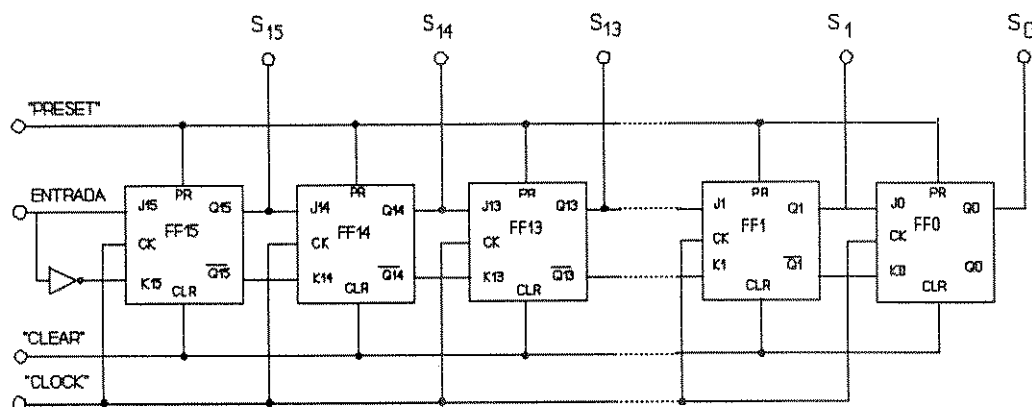


Fig. 5.8 Registrador de Deslocamento Série/Paralelo de 16 Bits

A configuração do flip-flop JK mestre-escravo utilizado está mostrada na Fig. 5.9. Após a montagem do dado para ser lido em paralelo pelo meio externo, há necessidade de que se mantenham os dados estáveis por um determinado período de tempo.

A necessidade acima pode ser atendida com o sinal HLD mostrado, pois enquanto estiver em 0, o escravo estará desabilitado, retendo a informação para que seja efetivada a leitura.

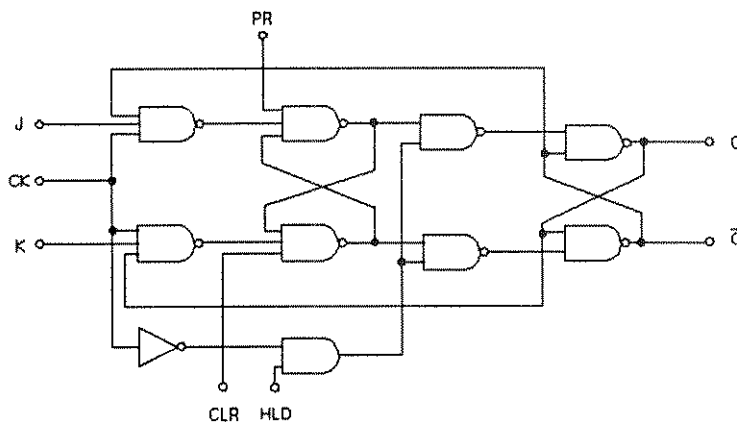


Fig. 5.9 Flip-Flop JK Mestre Escravo

O REGISTRADOR DE DESLOCAMENTO PARALELO-SÉRIE RDPS

A Fig. 5.10 mostra o diagrama em blocos do registrador de deslocamento paralelo/série de oito bits utilizado no multiplicador série e indicado como RDPS na Fig. 5.3.

Para que seja inicialmente zerado, é necessário fazer a linha de

"clear" baixa, bem como a de habilitação do "preset" e depois levá-la ao nível alto. Na sequência, um sinal 1 na entrada de "preset" fará com que a palavra de dado PR0/PR7 atinja os flip-flops. Assim, o pulso de habilitação do "preset" carrega todos os bits em paralelo no registrador.

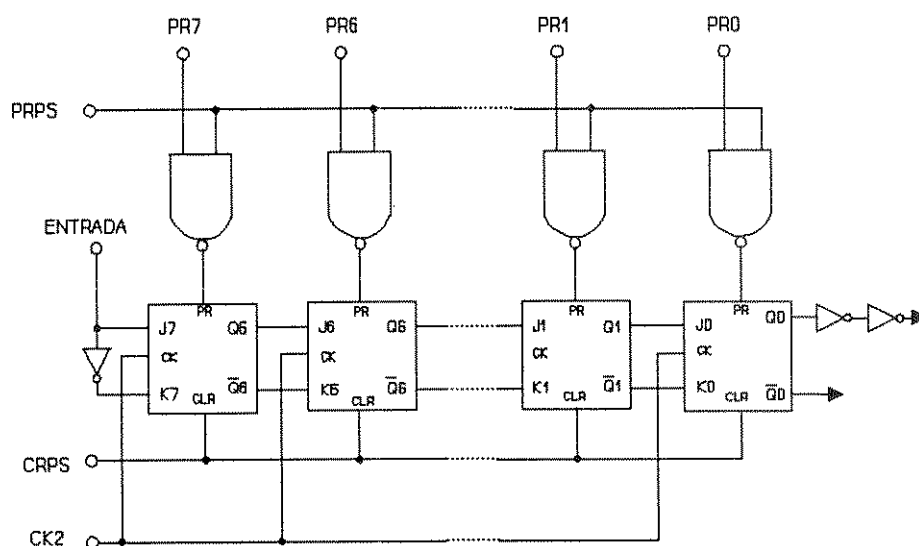


Fig. 5.10 Registrador Paralelo/Série de Oito Bits

A palavra armazenada pode agora ser lida serialmente pelo meio externo, um bit a cada pulso de "clock" CK, iniciando-se com o menos significativo.

Neste registrador, os flip-flops JK mestre-escravo não necessitam do sinal HLD utilizado na estrutura do registrador série/paralelo. Assim, a sua configuração pode ser simplificada para a mostrada na Fig. 5.11.

5.3.3 A UNIDADE DE CONTROLE

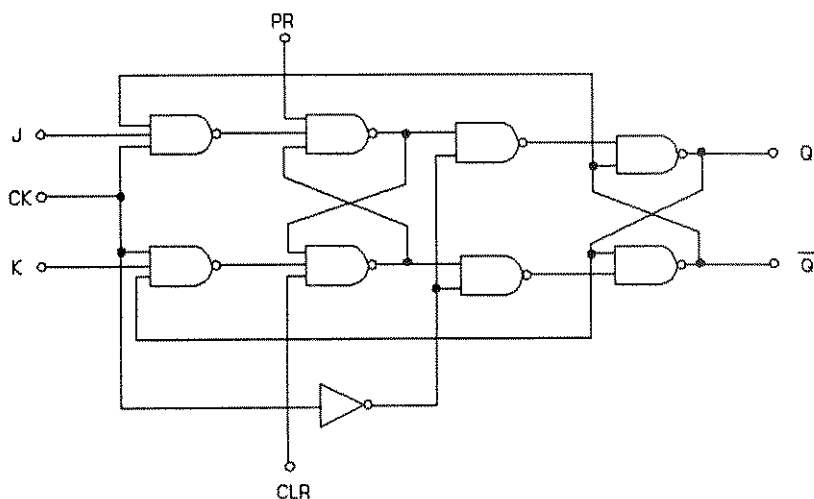


Fig. 5.11 Flip-Flop JK Mestre Escravo Simplificado

Como mostrado no diagrama em blocos do multiplicador série (Fig. 5.3), a Unidade de Controle recebe os sinais externos de temporização CK e CK2 bem como o sinal de início de multiplicação MULT e gera, a partir deles, os sinais PRPS e CRPS que envia aos registradores de deslocamento paralelo-série de oito bits.

A Unidade de Controle gera também os sinais de controle CNT e de "reset" RES, que envia às células básicas CELBA, bem como os sinais N20, PRSP e CRSP que aplica no registrador série-paralelo de dezesseis bits.

Finalmente, os sinais externos de "clock" CK e CK2 são repassados pela Unidade de Controle que também inverte estes dois sinais e os envia aos vários circuitos como CKBAR e CK2BAR.

O GERADOR DE FASES

Os sinais de "clock" CK e CK2, conforme mencionado acima, devem ser gerados externamente. Uma primeira análise mostraria que os sinais apresentados na Fig. 5.12 seriam bem adequados às exigências do projeto.

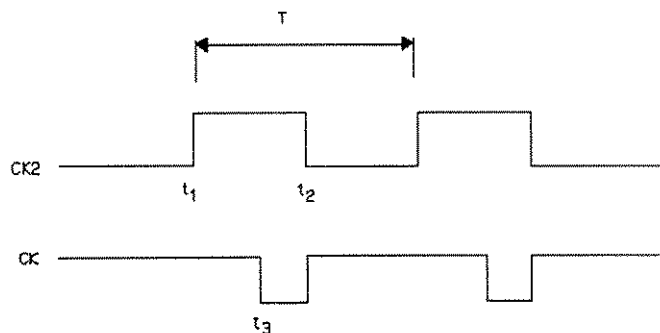


Fig. 5.12 Os Sinais Externos CK e CK2

OS SINAIS DE CONTROLE

A Fig. 5.13 mostra as formas de onda dos sinais de controle que a unidade deve enviar aos demais blocos do multiplicador de oito bits para que a computação possa ser efetivada com sucesso.

A GERAÇÃO DOS SINAIS PRPS, CRPS E RES

A Fig. 5.14 mostra o diagrama em blocos de um circuito que gera os sinais RES, CRPS e PRPS. O sinal indicado como PR-11 é o próprio sinal de habilitação

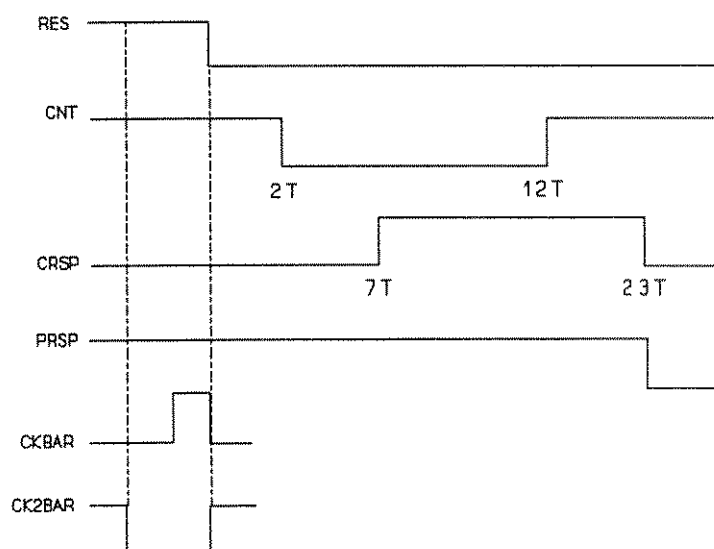


Fig. 5.13 Formas de Onda dos Sinais de Controle

externo MULT para início de multiplicação e deve ter uma duração menor que a menor largura de qualquer dos sinais a serem gerados. No caso em questão, o sinal que

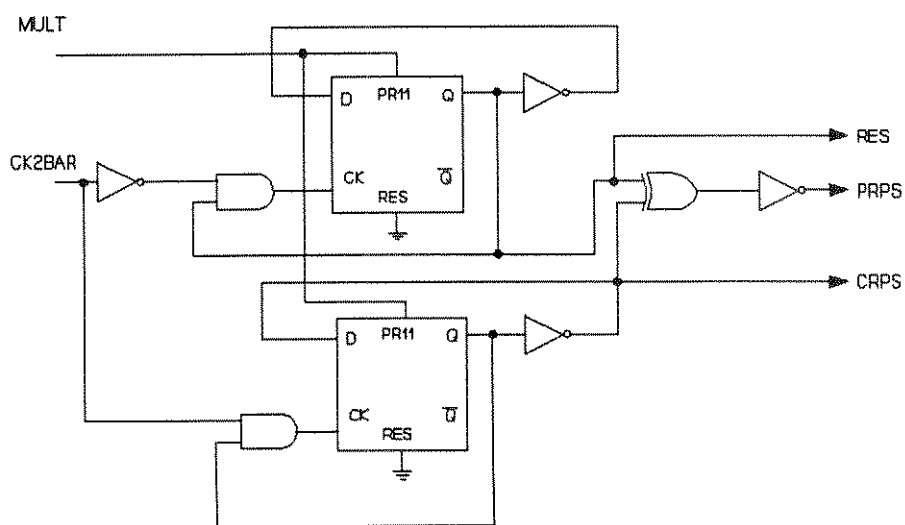


Fig. 5.14 Circuito de Geração dos Sinais RES, CRPS e PRPS

terá a menor largura é o PRPS de $0,5T$. Logo, a duração do sinal PR-11 deverá ser menor que $0,5T$.

Para a geração dos demais sinais, primeiro projetou-se um contador de 5 bits, como indicado na Fig. 5.15, que apresenta uma contagem em sequência binária normal até o equivalente decimal 23. Os vinte e quatro diferentes estados obtidos correspondem exatamente à temporização necessária para se realizar um ciclo completo de multiplicação.

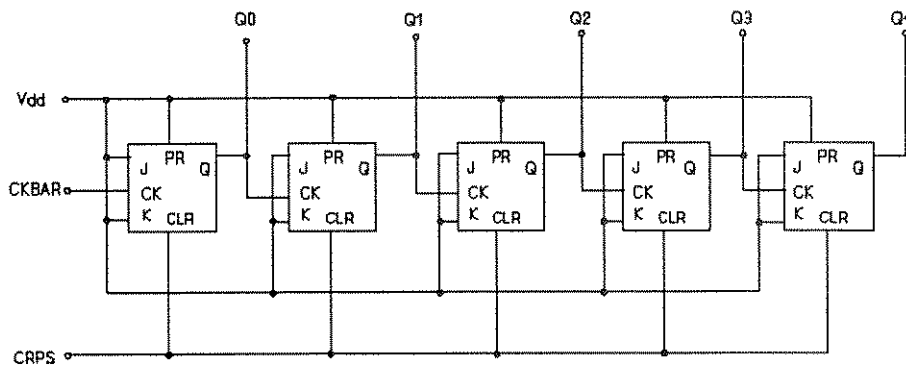


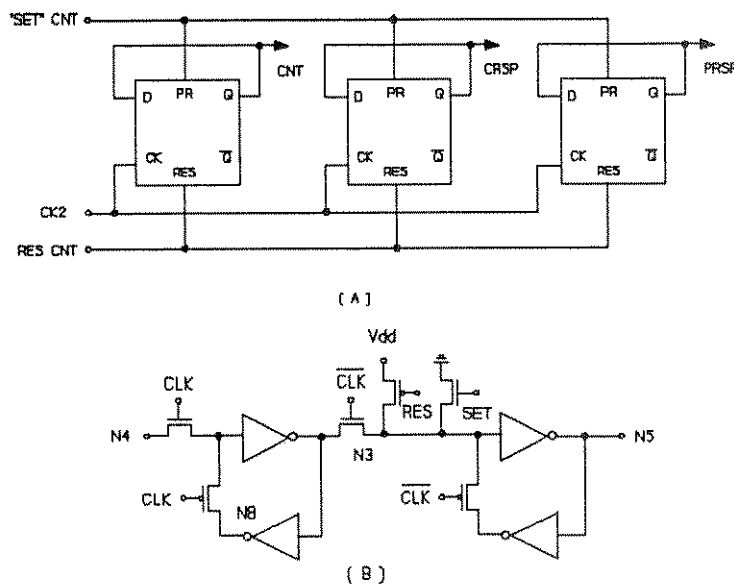
Fig. 5.15 Contador de Cinco Bits

No ciclo da multiplicação, considerando-se um período de "clock" T , a sequência executada será:

- Em $t=0$, deve-se propiciar um "set" nos sinais CNT e PRPS e um "reset" em CRSP. Entretanto, isto apenas ocorrerá se RES=1 pois este indica o início do processo de multiplicação.
- Em $t=2T$, deve-se dar um "reset" no sinal CNT.
- Em $t=7T$, deve-se dar um "set" no sinal CRSP.

- Em $t=12T$, deve-se dar um "set" em CNT.
- Em $t=23T$, deve-se dar um "reset" nos sinais CRSP e PRSP.

Para se atingirem as condições acima, utilizou-se flip-flops tipo D com "set" e "reset", conectados conforme a Fig. 5.16a. A arquitetura interna do flip-flop utilizado está ilustrada na Fig. 5.16b.



A lógica necessária para se fazer o "set" ou o "reset" de cada flip-flop D gerador dos sinais de controle será apresentada na sequência.

LÓGICA PARA "SET" E "RESET" DO SINAL CNT

A Tab. 5.2 mostra a lógica que deve ser executada pelos flip-flops D de forma a atender as exigências do sinal CNT.

T	Q ₄	Q ₃	Q ₂	Q ₁	Q ₀	Set	Reset
0	0	0	0	0	0	1	0
2T	0	0	0	1	0	0	1
12T	0	1	1	0	0	1	0

Tabela 5.2 Lógica de "Set" e "Reset" do Sinal CNT

Pode-se concluir que as expressões booleanas correspondentes aos sinais de "set" e de "reset" que devem ser aplicados nas respectivas entradas dos flip-flops D, para conformá-lo ao sinal CNT, são:

$$SET = \overline{Q_0} \overline{Q_1} \overline{Q_2} \overline{Q_3} \overline{Q_4} RES + \overline{Q_0} \overline{Q_1} Q_2 Q_3 \overline{Q_4}$$

ou,

$$SET = \overline{\overline{Q_0} \overline{Q_1} \overline{Q_2}} + \overline{\overline{Q_3} \overline{Q_4} RES} + \overline{\overline{Q_0} \overline{Q_1} Q_2} + \overline{\overline{Q_3} \overline{Q_4}} \quad (5-4)$$

e

$$RESET = \overline{Q_0} Q_1 \overline{Q_2} \overline{Q_3} \overline{Q_4}$$

ou,

$$RESET = \overline{\overline{Q_0} Q_1 \overline{Q_2}} + \overline{\overline{Q_3} \overline{Q_4}} \quad (5-5)$$

A Fig. 5.17 mostra o diagrama lógico do circuito que realiza as expressões booleanas dadas pelas Eqs. (4-4) e (4-5).

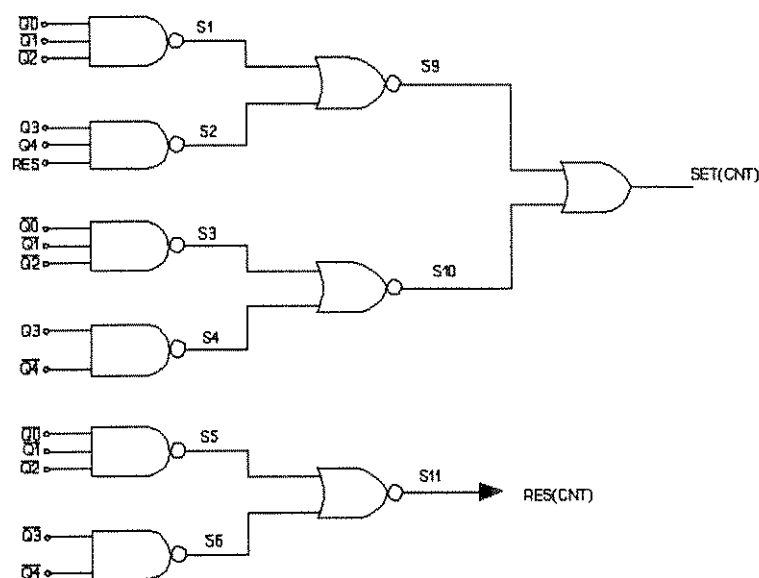


Fig. 5.17 Geração de "Set" e "Reset" do Sinal CNT

LÓGICA PARA "SET" e "RESET" DO SINAL CRSP

A Tab. 5.3 mostra a lógica que deve ser executada pelos flip-flops D de forma a atender ao sinal CRSP enviado pela Unidade de Controle ao registrador de deslocamento série/paralelo.

A tabela conduz às seguintes expressões booleanas dos sinais de "set" e de "reset" a serem aplicados aos flip-flops:

$$SET = Q_0 Q_1 Q_2 \overline{Q_3} \overline{Q_4}$$

ou,

T	Q ₄	Q ₃	Q ₂	Q ₁	Q ₀	Set	Reset
0	0	0	0	0	0	0	1
7T	0	0	1	1	1	1	0
23T	1	0	1	1	1	0	1

Tabela 5.3 Lógica de "Set" e "Reset" do Sinal CRSP

$$SET = \overline{\overline{Q_0} \overline{Q_1} \overline{Q_2}} + \overline{\overline{Q_3} \overline{Q_4}} \quad (5-6)$$

e,

$$RESET = \overline{Q_0} \overline{Q_1} \overline{Q_2} \overline{Q_3} \overline{Q_4} RES + Q_0 Q_1 Q_2 \overline{Q_3} \overline{Q_4}$$

ou,

$$RESET = \overline{\overline{Q_0} \overline{Q_1} \overline{Q_2}} + \overline{\overline{Q_3} \overline{Q_4} RES} + \overline{\overline{Q_0} \overline{Q_1} \overline{Q_2}} + \overline{\overline{Q_3} \overline{Q_4}} \quad (5-7)$$

A Fig. 5.18 mostra o diagrama lógico do circuito que implementa as funções descritas nas Eqs. (4-6) e (4-7).

LÓGICA PARA "SET" e "RESET" DO SINAL PRSP

A Tab. 5.4 mostra a lógica que deve ser executada pelos flip-flops de forma a atender as exigências do sinal PRSP .

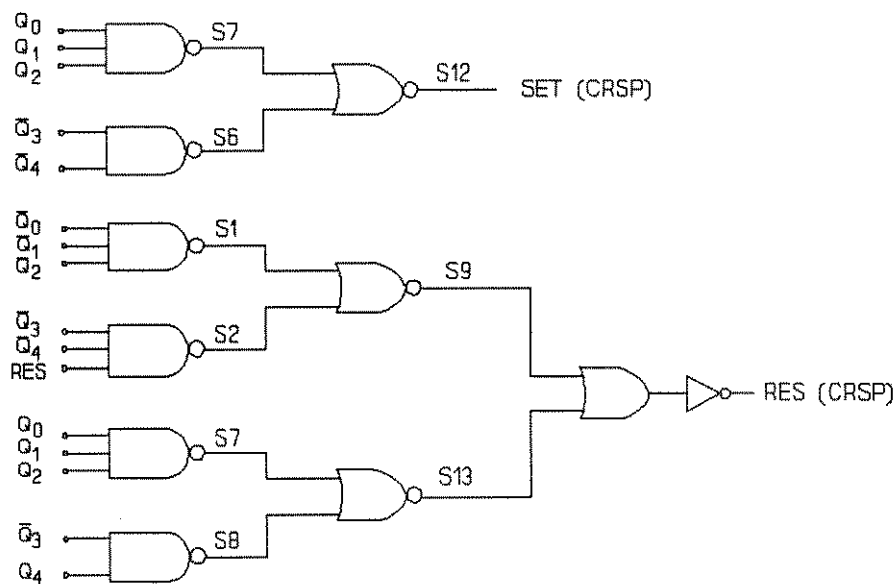


Fig. 5.18 Lógica de "Set" e "Reset" do Sinal CRSP

T	Q ₄	Q ₃	Q ₂	Q ₁	Q ₀	Set	Reset
0	0	0	0	0	0	1	0
23T	1	0	1	1	1	0	1

Tabela 5.4 Lógica de Set e Reset do Sinal PRSP

A tabela conduz às seguintes expressões booleanas para o circuito de geração dos sinais "set" e "reset" dos flip-flops:

$$SET = \overline{Q_0} \overline{Q_1} \overline{Q_2} \overline{Q_3} \overline{Q_4} RES$$

ou,

$$SET = \overline{\overline{Q_0} \overline{Q_1} \overline{Q_2}} + \overline{\overline{Q_3} \overline{Q_4} RES} \quad (5-8)$$

e,

$$RESET = Q_0 Q_1 Q_2 \overline{Q_3} Q_4$$

ou,

$$RESET = \overline{\overline{Q_0} \overline{Q_1} \overline{Q_2}} + \overline{\overline{Q_3} \overline{Q_4}} \quad (5-9)$$

As expressões booleanas acima conduzem aos circuitos lógicos mostrados na Fig. 5.19.

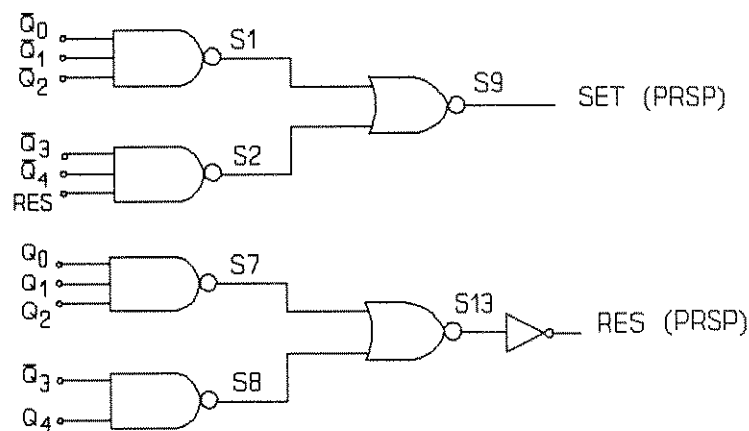


Fig. 5.19 Lógica de Geração de "Set e "Reset" do Sinal PRSP

5.4 A SIMULAÇÃO DO MULTIPLICADOR

5.4.1 SIMULAÇÃO ELÉTRICA

Os procedimentos adotados para a simulação elétrica do multiplicador série de oito bits são ligeiramente diferentes dos aplicados à simulação do multiplicador paralelo.

A estrutura em blocos independentes e o comportamento assíncrono do multiplicador paralelo permitiram uma simulação elétrica de cada bloco seguida de uma análise final de tempos pelo caminho crítico.

No caso do multiplicador série deste Capítulo, analisando-se a arquitetura exibida na Fig. 5.3, percebe-se tratar de um circuito síncrono em que o estudo de tempos devem ser restritos à determinação do período de "clock" mínimo necessário à operação mais rápida do multiplicador.

A exigência apontada acima fica evidenciada se se considerar o quanto é mais lento um multiplicador série que um paralelo. No caso do multiplicador série de oito bits, ter-se-á o resultado de uma multiplicação validado após a execução de vinte e três períodos completos de "clock".

Inicialmente, pensou-se em projetar um multiplicador que fosse temporizado externamente por um sinal de "clock" padrão com um ciclo de trabalho de 50%.

Uma análise posterior mostrou que o tempo de mestre dos flip-flops empregados, considerado como o tempo necessário para que um sinal se propague ao longo dos circuitos dos mestres e dos a ele conectados se estabilize na entrada dos circuitos dos escravos, era muito diferente que o tempo de escravo.

A afirmativa acima é óbvia se se considerar que o tempo de escravo dos flip-flops corresponde ao tempo necessário a que um sinal se propague ao longo do circuito do escravo somado ao tempo necessário para que este sinal saia do flip-flop e viaje pelo caminho mais crítico em tempo conectado aos mesmos e se estabilize, autorizando a realização do próximo período de "clock".

Com as considerações acima, a opção por um sinal de "clock" com 50% de ciclo de trabalho ficaria inviável pois tornaria o multiplicador extremamente lento, considerando que o mestre dos flip-flops não necessita do mesmo tempo do escravo.

A melhor solução encontrada foi a utilização do sinal de "clock" CK indicado na Fig. 5.12, onde o tempo t_2 - t_3 deve ser ajustado para permitir a operação dos circuitos mestre ou escravo dos flip-flops e o restante do ciclo, bem maior que este tempo, deve ser aplicado à operação dos circuitos mestres ou escravos e à propagação do sinal pelo circuitos que compõem o caminho mais crítico conectado a eles.

Embora apenas o sinal de "clock" CK pudesse atender às exigências de velocidade do multiplicador, há circuitos internos, como os registradores de deslocamento, em que um sinal de temporização do tipo CK2, permite uma operação correta, pois os tempos de mestre são quase idênticos aos tempos de

escravo. Por este motivo, preferiu-se trabalhar com os dois sinais, para que numa outra oportunidade se pudesse implementar um circuito transformando o sinal de "clock" CK2 no sinal CK.

Outra consideração válida para se justificar o trabalho com os dois sinais de temporização é o fato de que o sinal CK pode ser gerado internamente à partir do sinal CK2 e só não o foi por falta de área disponível no PMU. Com esta implementação, o sinal externo seria único e do tipo padrão com ciclo de trabalho igual a 50 %.

Para se prosseguir com a simulação elétrica, é importante que se determinem os tempos de propagação dos flip-flops D e JK utilizados no projeto. Na sequência, de posse destes elementos, pode se proceder a uma análise criteriosa buscando-se o caminho mais crítico dos circuitos conectados aos escravos para se estabelecer os maiores tempos de propagação e, conseqüentemente, o período de "clock".

O FLIP-FLOP D

A Fig. 5.6 mostra o diagrama em blocos do flip-flop D utilizado na célula básica do multiplicador. Entretanto, o flip-flop D utilizado na Unidade de Controle deve ser equipado com entradas de "set" e de "reset" instantâneas. Assim, a Fig. 5.20 mostra esta estrutura utilizada, que foi submetida ao simulador elétrico PSPICE.

A Tab. 5.5 mostra a temporização do mestre MST e do escravo SLV do

flip-flop D para o pior caso de velocidade. Embora o circuito do mestre e do escravo sejam semelhantes, a tabela indica um escravo mais lento. Na realidade, o circuito do

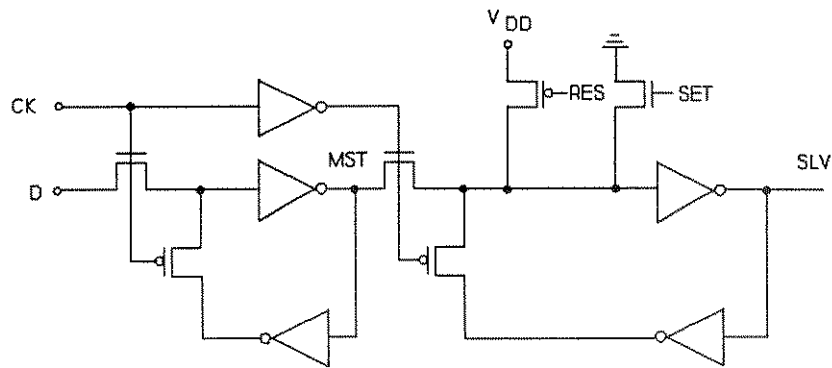


Fig. 5.20 Flip-Flop D com "Set" e "Reset" Instantâneos

deverá excitar outros circuitos que foram representados nesta simulação por uma capacitância de 0.05 pF, tornando-se mais lento que o mestre.

CASOS	Tempo de Propagação (nS)			
	Mestre		Escravo	
	Baixo para Alto	Alto para Baixo	Baixo para Alto	Alto para Baixo
TÍPICO	0.8	0.7	1.5	2.2
PIOR CASO DE VELOCIDADE	1.4	1.7	2.9	4.9
PIOR CASO DE POTÊNCIA	0.4	0.3	1.2	1.4

Tabela 5.5 Temporização do Flip-Flop D

O FLIP-FLOP JK

O flip-flop JK levado à simulação elétrica para a obtenção de sua temporização é o mostrado na Fig. 5.9. O sinal HLD é ativado pelo sinal PRSP enviado pela Unidade de Controle, forçando o registrador de deslocamento série / paralelo a manter nas suas saídas o resultado da multiplicação, até que o circuito externo possa fazer a leitura.

Outra consideração importante é que os sinais de "clear" e de "preset" somente podem ser ativados de forma síncrona, por estarem conectados ao mestre, ao contrário do flip-flop D que é ativado assincronicamente.

A Tab. 5.6 mostra a temporização do flip-flop JK onde os tempos de propagação foram tomados da forma mais crítica, ou seja, considerando sempre a saída mais lenta (Q ou a complementada) e no pior caso de velocidade.

CASOS	Tempo de Propagação (nS)			
	Mestre		Escravo	
	Baixo para Alto	Alto para Baixo	Baixo para Alto	Alto para Baixo
TÍPICO	1.8	2.5	2.6	3.9
PIOR CASO DE VELOCIDADE	2.7	4.5	4.9	7.3
PIOR CASO DE POTÊNCIA	1.4	1.9	1.3	2.2

Tabela 5.6 Temporização do Flip-Flop JK

O CAMINHO CRÍTICO

Após uma criteriosa análise do multiplicador série, concluiu-se que o caminho de tempo mais crítico corresponde ao da propagação do sinal pelo escravo do bloco D12 da Fig. 5.4, passando pelos blocos INV2, E3, OR1, XOR2, pelo somador da CELBA e pelo mestre do bloco D10. A Fig. 5.21 mostra este caminho em **negrito**.

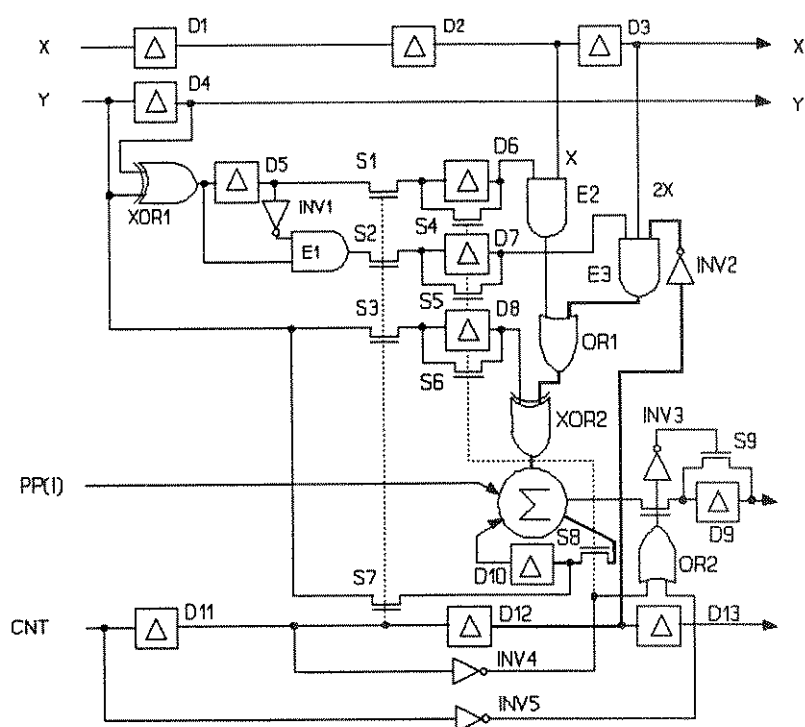


Fig. 5.21 Estabelecimento do Caminho Crítico

O período de "clock" CK indicado na Fig. 5.12 é formado por um tempo em nível baixo necessário à propagação do sinal pelos circuitos do escravo do flip-flop

D. Entretanto, este mesmo tempo também é utilizado para a propagação do sinal pelo mestre do flip-flop JK. Como este é mais lento, o tempo a ser considerado é o do flip-flop JK, com valor já simulado e igual a 4.5 nS no pior caso de velocidade.

Para se obter o restante do período de clock, ou seja, a parte em nível alto, utilizou-se o circuito da Fig. 5.22, desenhado com base na Fig. 5.21, submetendo-o à simulação elétrica. O resultado obtido, em pior caso de velocidade, foi de 12.0 nS, conforme indicado na Tab. 5.7, o que permite concluir que o período total ficará em torno de 16.5 nS.

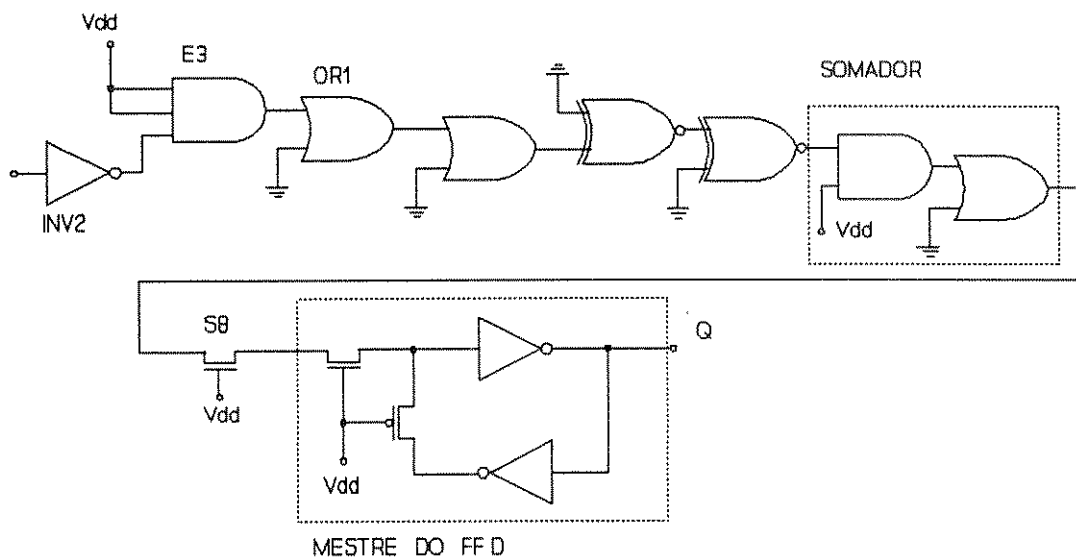


Fig. 5.22 O Caminho Crítico

Para o estabelecimento do sinal CK2, após determinado o período de "clock" de 16.5 nS e considerando a sua natureza de ciclo de trabalho igual a 50%, chegou-se aos valores de tempo em níveis alto e baixo iguais a 8.25 nS. Estes valores aplicados aos caminhos críticos estabelecidos por CK2, revelaram-se satisfatórios.

A Fig. 5.12 mostra os sinais de temporização mínimos CK e CK2 aplicáveis ao multiplicador série, onde o período T obtido é de 16.5 nS e o intervalo de tempo t_2-t_3 equivale a 4.5 nS.

Com os dados da tabela e considerando-se que são necessários vinte e três períodos de "clock", o tempo necessário para se conseguir o resultado de uma multiplicação completa é de 379.5 nS. Este valor permite que se trabalhe com velocidades de processamento em torno de 2.6 milhões de multiplicações por segundo.

CASOS	Tempo de Propagação (nS)	
	Sinal Q	
	Baixo para Alto	Alto para Baixo
TÍPICO	5.9	5.6
PIOR CASO DE VELOCIDADE	11.8	12.0
PIOR CASO DE POTÊNCIA	3.4	3.0

Tabela 5.7 Temporização do Caminho Crítico

5.4.2 A SIMULAÇÃO LÓGICA

O circuito do multiplicador paralelo, foi submetido ao Simulador Hilo 3M instalado em um computador VAX, embora a simulação elétrica já indicasse qualquer lógica incorreta. O volume dos resultados da simulação impede a sua anexação a este

trabalho, embora esteja disponível a eventuais interessados.

5.5 O "LAYOUT" DO MULTIPLICADOR

O "layout" do multiplicador série foi editado em conjunto com o do multiplicador paralelo para ser processado através do PMU CMOS6 em tecnologia de camada dupla de metal da ES2, utilizando-se os transistores em suas dimensões mínimas.

Assim como no caso do multiplicador paralelo, pretendia-se incluir, de forma separada, os vários blocos do multiplicador, assegurando uma boa condição para os testes a serem realizados durante a fase de caracterização.

A necessidade de se incluir o projeto dos dois multiplicadores em uma área bastante reduzida impossibilitou esta prática pois o multiplicador série ainda exigia a integração de 37 "pads". Assim, desenvolveu-se um "layout" em que cada célula básica ocupou uma área de $651,5 \times 247 \mu\text{m}$, o registrador de deslocamento série-paralelo $2.771 \times 127,5 \mu\text{m}$, o registrador paralelo-série $1.427 \times 171,5 \mu\text{m}$ e a unidade de controle, uma área de $1.531 \times 167 \mu\text{m}$.

Decidiu-se implementar o multiplicador série, assim como o paralelo, sem se criar mecanismos facilitadores de testes, embora contando com a estrutura dos laboratórios do CTI.

Com o intento de minimizar as áreas integradas, os transistores MOS de

canal n foram implementados com as dimensões mínimas do processo, ou seja, $L = 2\mu\text{m}$ e $W = 3\mu\text{m}$. Já os de canal p foram executados com $L = 2\mu\text{m}$ e $W = 6\mu\text{m}$ para compensar as diferentes mobilidades de portadores no canal.

Utilizando-se o editor gráfico KIC adaptado para microcomputadores, os "layouts" dos vários blocos foram implementados. Entretanto, a montagem final do circuito apenas pode ser realizada com o ferramental do CTI que também possibilitou que o resultado fosse submetido, com sucesso, ao DRC.

A Fig. 5.23 mostra uma "hardcopy" do "layout" final do multiplicador série que, infelizmente, não pode ser submetido a um software extrator de parâmetros pois o CTI não dispunha dessa ferramenta. O fato é grave pois eleva a possibilidade de ocorrência de conexões erradas que dificilmente poderiam ser descobertas por outros métodos.

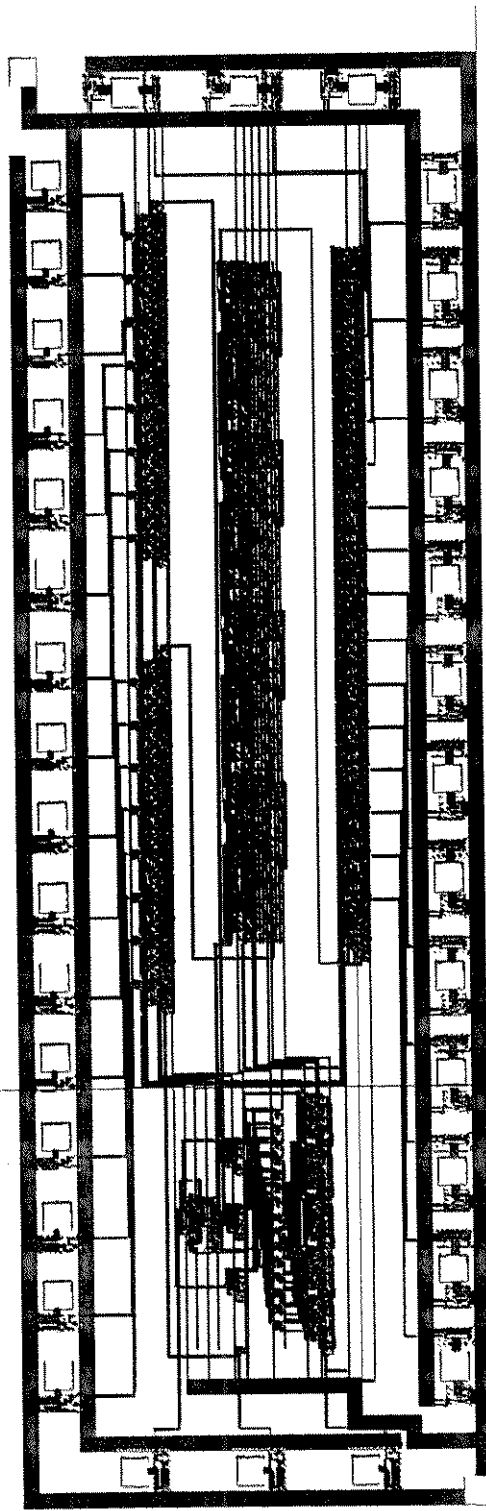
5.6 RESULTADOS E CONCLUSÕES

A Fig.5.24 é uma fotomicrografia deste circuito, difundido com o multiplicador paralelo no mesmo PMU, onde se pode perceber o cuidado em se ocupar pouca área e, ao mesmo tempo, manter uma distribuição de fácil identificação dos blocos visando facilitar os testes.

É importante ressaltar que este projeto, realizado para palavras de 8x8 bits mantém toda a sua modularidade para permitir estruturas de até 64 bits, limite do algoritmo de Booth.

SECRET

dy:8 width:1200 cell:mult.kic
 #m up:72 8 9 10 11 12

[illegible]

5.40

Os resultados obtidos nas simulações indicam que o bloco atende com sucesso as especificações previstas e que multiplicações à uma taxa de até 2,6 milhões por segundo podem ser atingidas.

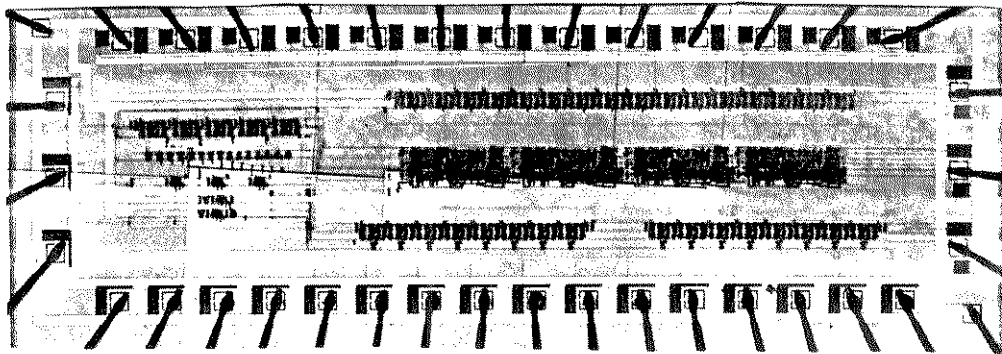


Fig. 5.24 Fotomicrografia do Multiplicador Série

Assim como no caso do multiplicador paralelo, foram detetados erros de "layout" e a inexistência de equipamentos atuais de diagnóstico impediram que resultados confiáveis pudessem ser agregados a este trabalho.

CAPÍTULO 6

CONCLUSÕES FINAIS

A proposição inicial de escolha dos três blocos descritos nos capítulos anteriores para integrarem este trabalho revelou-se bastante precisa pois as simulações e os testes indicaram uma alta performance, apropriados à integração em complexos circuitos VLSI e ainda, caracterizam-se como circuitos de alta repetição dentro dos mais variados sistemas digitais.

O conversor A/D projetado de 8 bits com a estrutura proposta, permite que se consiga taxas de amostragens superiores a 1 MHz e as simulações indicam que, com uma pequena melhoria do circuito de auto-zero dos comparadores, frequências de até 50 MHz podem ser conseguidas sem qualquer complexidade.

No caso de conversor acredita-se que, para propósitos de trabalhos

semelhantes ao aqui desenvolvido, não deve ser tentada a solução tipo "flash" convencional. Esta solução implica em grandes áreas de integração e acaba se traduzindo no próprio circuito VLSI e não em um bloco a ser utilizado em um circuito VLSI.

A solução "flash" modificada proposta não é única e abre uma grande área de pesquisa de outras soluções comprometidas com a busca da menor área de integração possível em função da maior taxa de amostragem.

A estrutura do multiplicador paralelo projetado pode permitir frequências de multiplicação de até 35 MHz, trabalhando-se com palavras binárias de 4 bits. A excelente performance do circuito "carry look-ahead" adicionado ao bloco, faz com que para palavras maiores de 4 bits, a redução da frequência de multiplicação não seja sensível. Por outro lado, quando as palavras são bem maiores, abre-se um interessante estudo de compromisso entre a menor área de integração e o correto dimensionamento do somador "carry look-ahead" que basicamente representa a frequência de operação.

Já o multiplicador série conduziu a velocidades de operação de até 2,6 milhões de multiplicações por segundo, trabalhando com palavras de 8 bits. Para um multiplicador série, que usualmente é lento, a estrutura escolhida revelou-se de comportamento excelente.

Os blocos desenvolvidos neste trabalho, embora com os previstos "tropeços" no "layout" editado no pré-histórico "KIC", encontram-se disponíveis a todos que dispuserem de um software gráfico mais eficaz. O sucesso da realização é assegurado pelos resultados obtidos nas simulações.

Acredita-se ser importante registrar que o sucesso do trabalho como um todo foi decisivo na abertura da área de microeletrônica na EFEI. Hoje, a equipe deste setor encontra-se desenvolvendo conversores A/D após ter dominado a técnica de chaveamento de corrente em conjunto com dois grupos europeus.

Outro aspecto que se acredita importante registrar é que a não periodicidade dos PMUs leva os grupos acadêmicos a apresentar projetos já globais e mesmo que tal procedimento impossibilite a inclusão de estruturas de teste. O preço a ser pago é a dificuldade de caracterização dos seus circuitos.

Os grupos mais ousados que tentarem o desenvolvimento de circuitos integrados mais complexos, correm o risco de não terem como testá-los. Hoje ainda diagnostica-se prováveis defeitos à base de decapagem química comum ou com plasma, como a que se realizou neste trabalho. O comprometimento do comportamento elétrico dos componentes pelo ataque químico não assegura um bom diagnóstico, não é confiável, é demorado e pode até destruir as amostras sem que se obtenha qualquer informação conclusiva.

Lastimavelmente, sabe-se que qualquer grupo pertencente a países que investem na área, pode dispor de equipamentos de diagnóstico de falhas de circuitos integrados compatíveis à complexidade de seus circuitos. Como ilustração poder-se-ia mencionar os equipamentos que propiciam o acesso direto de trilhas utilizando "electron-beam" à partir do diagrama esquemático e do "layout" [19] e [20].

A conclusão deste trabalho coloca o autor junto aos poucos visionários e missionários da microeletrônica brasileira. Embora comprovadamente a área seja estratégica e reconhecida desta forma junto aos órgãos governamentais, apenas ações

tímidas e mal direcionadas são tentadas na redução do "gap" tecnológico que esta área apresenta em relação aos países mais desenvolvidos. Nenhum grupo acadêmico encontra-se equipado de forma a poder resolver seus projetos de forma confiável, rápida e competitiva. Os reflexos são dirigidos a todo o parque industrial brasileiro que tem o seu desenvolvimento constantemente ameaçado pela impossibilidade de competir em igualdade.

Sem uma política adequada à microeletrônica e comprometimento de todos os segmentos de ciência e tecnologia brasileiros, pode-se aqui, no limiar do novo milênio, o País voltar a ser subjugado não mais pela força mas pela imposição da pior forma de escravidão: a tecnológica.

BIBLIOGRAFIA REFERENCIADA

- [1] Booth, A.D. - **"A Signed Binary Multiplication Technique"** - Q.J.Mech.Appl.Math.4
- pp 236/240 - 1.951.
- [2] Booth, A.D. and Booth, K.H.V.- **"Automatic Digital Calculator"** - Zend ed. New York
Academic - 1.956.
- [3] Baugh, C.R. and Wooley, B.A. - **"A Two's Complement Parallel Array
Multiplication Algorithm"** - IEEE Trans. on Comp. C22 - pp 1.045/1.047 - 1.973.
- [4] Cappello, P.R. and Steiglitz, K. - **"A VLSI Layout for a Pipelined Dadda
Multiplier"** - ACM Trans. on Computer Systems 1(2) - pp 157/174 - May 1.983.
- [5] Luk, W.K. - **"A Regular Layout for Parallel Multiplication of $O(\log^2 n)$ Time"**. In
Kung, H.T.; R.F.Sproull and G.L.Steele (editor), "VLSI Systems and Computations",
pp 317/326. Computer Science Department, Carnegie-Mellon University, Computer
Science Press, Inc.
- [6] West, N. and Eshraghian, K. - **"Principles of CMOS VLSI Design - A System
Perspective"** - Addison Wesley Publishing Company - 1.988.

- [7] Preparata, F.P. - **"A Mesh-Connected Area-Time Optimal VLSI Integer Multiplier"** - In Kung, H.T.; R.F. Sproull and G.L. Steele (editors), **VLSI Systems and Computations** - pp 311/316. Computer Science Department, Carnegie-Mellon University, Computer Science Press, Inc. - October 1.981.
- [8] Garner, H.L. - **"The Residue Number System"** - IEEE Trans. on Electronics Computers - pp 140-147 - June 1.959.
- [9] Y. Chu - **"Digital Computer Design Fundamentals"** - New York - McGraw-Hill Book Company - 1.962.
- [10] Cavanagh, J.J.F. - **"Computer Science Series : Digital Computer Arithmetic"** - McGraw-Hill Book Company - 1.984.
- [11] Annaratone, M. and Shen, W.Z. - **"The Design of a Booth Multiplier: nMos vs. CMOS Technology"** - Proc. of the 2nd International Symposium on VLSI Technology, Systems and Applications - Taiwan - 1.985.
- [12] Masumoto, R.T. - **"The Design of a 16x16 Multiplier"** - VLSI Design (Lambda) First Quarter - pp 15/21 - 1.980.
- [13] Ware, F.A. et al. - **"64 Bit Monolithic Floating Point Processors"** - IEEE Journal of Solid State Circuits SC-17(5) - pp 898/907 - Oct - 1.982.
- [14] Jackson, L.B.; Kaiser, S.F. and McDonald, H.S. - **"An Approach to the Implementation of Digital Filters"** - IEEE Trans. Audio and Electroacoust., Vol AU-16, pp 413/421 - Sept 1.968.

- [15] Freeny, S.L.; Kiebertz, R.B.; Mina, D.V. and Tewksbury, S.K. - **"Design of Digital Filters for an all Digital TDM-FDM Translator"** - IEEE Trans. Circuit Theory, Vol. CT-18 - pp 702/711 - Nov 1.971.
- [16] Spinks, A.H. - **"2 KHz Modulator of the Digital A-Channel Bank"** - July 1970.
- [17] Lyon, R.F. - **"Two's Complement Pipeline Multipliers"** - IEEE Trans. on Communications - pp 418/425 - April 1.976.
- [18] Geiger, R.L.; Allen, P.E. and Strader, N.R. - **"VLSI Design Techniques for Analog and Digital Circuits"** - McGraw Hill, Inc. - pp 495/518 - 1.990.
- [19] Gregorian, R. and Temes, G.C. - **"Analog Integrated Circuits for Signal Processing"** - John Wiley & Sons - pp 425/436 - 1.986.
- [20] Haskard, M.R. and May, I.C. - **"Analog VLSI Design"** - Prentice Hall - pp 113/141 - 1.988.
- [21] Shen, B.J. - **"Switched-Induced Error Voltage on a Switched Capacitor"** - IEEE Journal on Solid-State Circuits - Vol. SC-19 - Nº 4 - Aug 1.984.
- [22] **Microelectronics Monitor** - 1.993.
- [23] Talbot, C.G. - **"The Progress in E-Beam Probing Technology"** - Evaluation Engineering - July 1.990.
- [24] Harari, S.L. and Talbot, C.G. - **"Scanning Electron Microscopes - Chip Diagnosis:**

A New Methodology" - Evaluation Engineering - Oct 1.987.

BIBLIOGRAFIA RECOMENDADA

Dingwall, Andrew G.F. - " **Monolithic Expandable 6B 15 MHz CMOS/SOS A/D Converter**" - IEEE International Solid State Circuits Conference - pp 126/127 - 1.979.

Demler, Michael J. - "Understand CMOS Flash ADC to Apply Them Effectively " EDN, pp 127/134 - Jan 1.988.

Costlow, Terry - "Flash 8-Bit Converter Samples at 150 MHz" - Electronic Design - pp 42/44 - Oct 1.986.

Ahmed, R.E. - "New Design Technique for Two-Step Flash A/D Converter" - Electronic Letters - Vol 26 - Nº 15 - pp 1185/1187 - July 1.990.

Dingwall, Andrew G.F. - "Monolithic Expandable 6 Bit 20 MHz CMOS/SOS A/D Converter" - IEEE Journal of Solid State Circuits - Vol SC-14 - Nº 6 - pp 926/932 - Dec 1.979.

Schreier, Paul G. - " IC Data Converters" - EDN - pp 73/86 - Aug 1.978.

Hornak, T. and Corcoran, J.J. - "A High Precision Component-Tolerant A/D Converter" - IEEE Journal of Solid-State Circuits" - Vol. SC-10, Nº 6, pp 386/391 - Dec 1.975.

Rodgers, B.J. and Thurber, C.R. - **"A Monolithic +/- 5 1/2 - Digit BIMOS A/D Converter"** - IEEE Journal of Solid State Circuits" - Vol 24 - N° 3 - pp 617/626 - June 1.989.

Tsukada, T.; Takagi, K. and Kita, Y. - **"An Automatic Error Cancellation Technique for Higher Accuracy A/D Converters"** - IEEE Journal of Solid-State Circuits" - Vol. SC-19 - N° 2 - pp 266/268 - April 1.984.

Van der Grift, Rob E.J.; Rutten, Ivo W.J.M. and Van der Veen, M. - **"An 8-Bit Video ADC Incorporating Folding and Interpolation Techniques"** - IEEE Journal of Solid-State Circuits - Vol. SC-22 - N° 6 - pp 944/953 - Dec 1.987.

Song, Bang-Sup; Lee, Seung-Hoon and Tompsett, M.F. - **"A 10-B 15 MHz CMOS Recycling Two-Step A/D Converter"** - IEEE Journal of Solid-State Circuits - Vol. 25 - N° 6 - pp 1328/1338 - Dec 1990.

Timko, M.P. and Holloway, P.R. - **"Circuit Techniques for Achieving High Speed-High Resolution A/D Conversion"** - Vol SC-15 - N° 6 - pp 1040/1051 - Dec 1.980.

Kumamoto, T.; Naraya, M. and Honda, H. - **"An 8-Bit High-Speed CMOS A/D Converter"** - IEEE Journal of Solid State Circuits - Vol. SC-21 - N° 6 - pp 976/982 - Dec 1.986.

Nairn, D.G. and Salama, A.T. - **"A Ratio-Independent Algorithmic Analog-to-Digital Converter Combining Current"** - IEEE Transactions on Circuits and Systems" - Vol. 37 - N° 3 - pp 319/325 - Mar 1.990.

Petschacher, R. and Zojer, B. - **"A 10-B 75-MSPS Subranging A/D Converter with**

Integrated Sample and Hold" - IEEE Journal of Solid State Circuits - Vol.25 - Nº 6 - pp 1339/1346 - Dec 1.990.

Mayes,M.K. and Chin, S.W. - **"A Multistep A/D Converter Family with Efficient Architecture"** - IEEE Journal of Solid-State Circuits - Vol 24 - Nº 6 - pp 1492/1498 - Dec 1.989.

Yukawa, A. - **"A CMOS 8-Bit High-Speed A/D Converter IC"** - IEEE Journal of Solid-State Circuits - Vol SC-20 - Nº 3 - pp 775/779 - June 1.985.

Fotovhi, B. and Hodges, D.A. - **"High-Resolution A/D Conversion in MOS/LSI"** - IEEE Journal of Solid-State Circuits - Vol SC-14 - Nº 6 - pp 920/925 - Dec 1.979.

Robert, J.; Temes, G.C. and Valencic, V. - **"A 16-Bit Low-Voltage CMOS A/D Converter"** - IEEE Journal of Solid-State Circuits - Vol SC-22 - Nº 2 - pp 157/163 - April 1.987.

Hosotani, S.; Miki, T. and Maeda, A. - **"An 8-Bit 20-MS/s CMOS A/D Converter with 50 mW Power Consumption"** - IEEE Journal of Solid-State Circuits - Vol 25 - Nº 1 - pp 167/172 - Feb 1.990.

Doernberg, J.; Gray, P.R. and Hodges,D.A. - **"A 10-Bit 5-MSample/s CMOS Two-Step Flash ADC"** - IEEE Journal of Soli State Circuits - Vol 24 - Nº 2 - pp 241/249 - April 1.989.

Kerth, D.A. and Sooch. N.S. - **"A 12-Bit 1-MHz Two-Step Flash ADC"** - IEEE Journal of Solid-State Circuits - Vol SC-19 - Nº 6 - pp 813/819 - Dec 1.984.

Hotta, M.; Maio, K. and Yokozawa, N. - **"A 150-mW 8-Bit Video-Frequency A/D Converter"** - IEEE Journal of Solid-State Circuits - Vol. SC-21, Nº 2 - pp 318/323 - April 1.986.

Van der Plassche, R.J. and Baltus, P. - **"An 8-Bit 100-Mhz Full-Nyquist Analog-to-Digital Converter"** - IEEE Journal of Solid State Circuits - Vol 23 - Nº 6 - pp 1334/1334 - Dec 1.988.

Zojer, B. and Petschacher, R. - **"A 6-Bit/200-Mhz Full Nyquist A/D Converter"** - IEEE Journal of Solid-State Circuits - Vol SC-20 - Nº 3 - pp 780/786 - June 1.985.

Bacrania, K. - **"A 12-Bit Successive-Approximation-Type ADC with Digital Error Correction"** - IEEE Journal of Solid-State Circuits - Vol SC-21 - Nº 6 - pp 1016/1025 - Dec 1.986.

Post, H.U. and Waldschmidt, K - **"A High-Speed NMOS A/D Converter with a Current Source Array"** - IEEE Journal of Solid-State Circuits - Vol SC-15 - Nº 3 - pp 295/301 - June 1.980.

Robert, J. and Deval, P - **"A Second-Order High-Resolution Incremental A/D Converter with Offset and Charge Injection Compensation"** - IEEE Journal of Solid-State Circuits - Vol 23 - Nº 3 - pp 736/741 - June 1.989.

Fernandes, J.; Lewis, S.R.; Mallinson, A.M. and Miller, G.A. - **"A 14-Bit 10 ms Subranging A/D Converter with S/H"** - IEEE Journal of Solid-State Circuits - Vol 23 - Nº 6 - pp 1309/1315 - Dec 1.988.

Redfern, T.P.; Jr, J.J.C. and Frederiksen, T.M. - **"A Monolithic Charge-Balancing Successive Approximation A/D Technique"** - IEEE Journal of Solid-State Circuits - Vol SC-14 - N° 6 - pp 912/919 - Dec 1.979.

Ishikawa, M. and Tsukahara, T. - **"An 8b 40 MHz Subranging ADC with Pipelined Wideband S/H"** - IEEE International Solid-State Circuits Conference - pp 12/13 - Feb 1.989.

Jaeger, R. C. - **"Tutorial: Analog Data Acquisition Technology - Part II - Analog-to-Digital Conversion"** - IEEE Micro - pp 46/56 - Aug 1.982.

Kutsuwa, T. and Ebata, K. - **"Configuration and Evaluation of 2's Complement Multiplication - Division Arrays"** - IEEE Transactions on Circuits and Systems - Vol.Cas 34 - N° 3 - pp 304/308 - March 1.987.

Donthi, R.V. and Singh, H. - **"Bit Sequential Multiplier Using Carry-Look-Ahead Technique"** - Int. J. Electronics - Vol.55 - n° 3 - pp 411/416 - 1.983.

Wey, C.L. and Chang, T.Y. - **"Design and Analysis of VLSI-Based Parallel Multiplier"** - IEE Proceedings - Vol. 137 - Pt.E - N° 4 - pp 328/336 - July 1.990.

Lee, J.Y.; Garvin, H.L. and Slayman, C.W. - **"A High-Speed High-Density Silicon 8x8-Bit Parallel Multiplier"** - IEEE Journal of Solid-State Circuits - Vol SC-22 - N° 1 - pp 35/40 - Febr 1.987.

Bakhtiarov, G.D. and Dzardanov, P.A. - **"State of the Art and Prospects for Development of High-Speed Analog-to-Digital Converters (Review)"** - Instruments

and Experience Techniques - 25(6) - pp 1315/1332 - 1.982.

Hatamian, M. and Cash, G.L. - **"A 70-MHz 8-BitX8-Bit Parallel Pipelined Multiplier in 2.5- μ M CMOS"** - IEEE Journal of Solid-State Circuits - Vol SC-21 - N° 4 - pp 505/513 - Aug 1.986.

Cooper, A.R. - **"Parallel Architecture Modified Booth Multiplier"** - IEE Proceedings - Vol 135 - Pt. G - N° 3 - pp 125/128 - June 1.988.

Matteme, L.; Van Meerbergen, J.; Beenker, F.; Mehra, V.; Theunissen, J. and Segers, R. - **"A C-Testable Booth Multiplier Designed for a Silicon Compilation Environment"** - Proceedings of the 1987 - IEEE Int. Conf. on Computers Design - USA - pp 354/357 - Oct 1.987.

Henlin, D.A.; Fertsch, M.T.; Mazin, M. and Lewis, E.T. - **"A 16 Bit X 16 Bit Pipelined Multiplier Macrocell"** - IEEE Journal of Solid State Circuits - Vol SC-20 - N° 2 - pp 542/547 - April 1.985.

Dadda, L. - **"On Parallel Digital Multipliers"** - Copyright 1.976 by Associazione Elettrotecnica ed Electronica Italiana - pp 126/132 - 1.976.

Dadda, L. - **"Some Schemes for Parallel Multipliers"** - Colloque sur L'Algèbre de Boole -Grenoble - 11 a 17 january 1.965.